

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Analýza a implementace nástroje pro sledování produktů**

## **Analysis and Implementation of the Product Tracking Tool**

# Zadání bakalářské práce

Student:

**Martin Jagoš**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Analýza a implementace nástroje pro sledování produktů**  
**Analysis and Implementation of the Product Tracking Tool**

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je implementovat uživatelský nástroj pro sledování produktů (vývoje, ceny, dostupnosti akce, letáky atd.) jako samostatnou webovou aplikaci. Práce bude zaměřena na inovativnost a zkoumání stávajících řešení, hledání nových modelů a přístupů v dané oblasti a jejich praktická implementace.

1. Student provede analýzu existujících nástrojů pro sledování produktových nabídek a to jak letákového typu, tak produktových katalogů (českých i zahraničních). Student analyzuje existující modely z pohledu implementace, obchodní strategie, zpoplatněných služeb a doplňkových funkcí.
2. Student provede návrh vlastního konceptu nástroje pro sledování produktů v podobě generického modelu pro libovolný obsah.
3. Student provede analýzu, návrh a implementaci nástroje pro sledování produktů ve vhodně zvoleném implementačním prostředí s použitím SQL nebo no-SQL databázových technologií.
4. V rámci implementace se student zaměří na pokročilé funkce vizualizace prezentovaného obsahu, responsivní design a další.
5. Student provede srovnání vlastního řešení s existujícími projekty.

Seznam doporučené odborné literatury:


- [1] GORMLEY, Clinton a Zachary TONG. Elasticsearch: the definitive guide. ISBN 1449358543.
- [2] SHKLAR, Leon. a Rich. ROSEN. Web application architecture: principles, protocols and practices. 2nd ed. Hoboken, NJ: Wiley, c2009. ISBN 047051860x.
- [3] SHIVAKUMAR, Shailesh Kumar. Architecting high performing, scalable and available enterprise web applications. ISBN 9780128022580.
- [4] WEERAWARANA, Sanjiva. Web services platform architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and more. Upper Saddle River, NJ: Prentice Hall PTR, c2005. ISBN 0131488740.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019

  
\_\_\_\_\_  
doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



  
\_\_\_\_\_  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

.....  
Joges

Rád bych poděkoval Ing. Radoslavu Fasugovi, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce.

## **Abstrakt**

Tato bakalářská práce se zaměřuje na analýzu a implementaci nástroje, který sleduje produkty v internetových obchodech z hlediska ceny a dostupnosti. Získaná data o produktech prezentuje uživatelům formou grafů a tabulek a tyto data dále zpracovává a používá je pro poskytování dalších funkcí nástroje. Cílem této práce je analyzovat existující nástroje pro sledování produktů z hlediska poskytovaných služeb, cenové dostupnosti a uživatelského hodnocení a implementovat tento nástroj pro sledování produktů pro uživatelem definované obchody. Obsahem této práce je analýza existujících řešení pro český a zahraniční trh, popis návrhu a implementace vlastního nástroje a výsledky jeho testování s možnými návrhy na vylepšení a doplnění novými funkcemi.

**Klíčová slova:** Sledování produktů, ASP.NET

## **Abstract**

This bachelor thesis focuses on analysis and implementation of tool which tracks prices and availability of products in shops on the internet. Gathered data about products are presented to users in form of charts and tables and these data are further processed and used for providing other functions of this tool. The goal of this thesis is to analyse existing product tracking tools in terms of provided services, price availability and user ratings and implement product tracking tool for user defined shops. This thesis contains analysis of existing solutions for czech and foreign market, design and implementation of own tool and the results of his testing with possible improvements and new functions.

**Key Words:** Product tracking, ASP.NET

# Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam tabulek	10
Seznam výpisů zdrojového kódu	11
<b>1 Úvod</b>	<b>12</b>
<b>2 Popis existujících řešení</b>	<b>13</b>
2.1 Zahraníční nástroje . . . . .	13
2.2 České nástroje . . . . .	15
<b>3 Popis zadání</b>	<b>17</b>
3.1 Obecný popis nástrojů . . . . .	17
3.2 Popis vlastního řešení . . . . .	17
<b>4 Datová a funkční analýza</b>	<b>18</b>
4.1 Datová analýza . . . . .	18
4.2 Funkční analýza . . . . .	23
4.3 Use case diagram . . . . .	38
<b>5 Návrh implementace</b>	<b>40</b>
5.1 Technologie pro implementaci . . . . .	40
5.2 Použité balíčky a knihovny . . . . .	40
5.3 Uživatelské rozhraní . . . . .	41
<b>6 Implementace</b>	<b>47</b>
6.1 ASP.NET MVC . . . . .	47
6.2 Datová vrstva . . . . .	48
6.3 Implementace uživatelského rozhraní . . . . .	49
<b>7 Testování</b>	<b>51</b>
<b>8 Návrhy na zlepšení</b>	<b>53</b>
<b>9 Závěr</b>	<b>55</b>
<b>Literatura</b>	<b>56</b>

## Seznam použitých zkratek a symbolů

DI	– Dependency Injection
MVC	– Model-View-Controller
URL	– Uniform resource locator
HTML	– Hypertext markup language
XML	– Extensible markup language
API	– Application programming interface
UML	– Unified modeling language
SQL	– Structured query language
CSS	– Cascading style sheets
DTO	– Data transfer object
DAO	– Data access object
ORM	– Objektově relační mapování
CSV	– Comma-separated values



## Seznam obrázků

1	Relační model databáze . . . . .	22
2	Use case diagram . . . . .	38
3	Formulář pro registraci uživatele . . . . .	41
4	Formulář pro přihlášení uživatele . . . . .	42
5	Stránka Můj účet . . . . .	42
6	Seznam produktů . . . . .	43
7	Historie produktu . . . . .	43
8	Formulář přidání produktu . . . . .	44
9	Registrace e-shopu . . . . .	44
10	Moje e-shopy . . . . .	45
11	Porovnání cen . . . . .	45
12	Screenshot uživatelského rozhraní aplikace . . . . .	50
13	Graf vývoje ceny produktu <i>Zlatá investiční mince Maple Leaf 1 Oz</i> . . . . .	51

## Seznam tabulek

1	Datový slovník tabulky uživatel . . . . .	18
2	Datový slovník tabulky product . . . . .	19
3	Datový slovník tabulky eshop . . . . .	19
4	Datový slovník tabulky user_eshop . . . . .	20
5	Datový slovník tabulky tracked . . . . .	20
6	Datový slovník tabulky notify . . . . .	21
7	Datový slovník tabulky product_cat . . . . .	21
8	Datový slovník tabulky product_history . . . . .	22

## Seznam výpisů zdrojového kódu

1	Ukázka akce controlleru Products . . . . .	47
2	Ukázka DTO <i>user</i> . . . . .	48
3	Ukázka metody z DAO <i>product</i> . . . . .	48

# 1 Úvod

Zřejmě první otázka, která čtenáře této bakalářské práce bude zajímat, je k čemu tento nástroj slouží. Jednotlivý provozovatelé internetových obchodů chtějí znát informace o nabízených produktech, jejich cenách, slevových akcích a dostupnosti produktů u svých konkurentů, aby mohli na jejich slevové akce a prodejní metody, jako jsou například akce 2+1 apod., reagovat a nabídnout tak nakupujícím lepší nabídku. K dosažení tohoto cíle mohou využít různé internetové srovnávače, procházet sociální sítě a různá diskuzní fóra, nebo navštěvovat přímo webové stránky e-shopů a manuálně shromažďovat data o produktech a analyzovat jejich nabídku. Na trhu se stovkou různých obchodů s daným sortimentem zboží se jedná o nadlidský úkol. Z tohoto důvodu existují nástroje pro sledování produktů, které výše zmíněné úkony automatizují a získaná data o produktech prezentují uživatelům v přívětivé podobě ve formě tabulek a grafů.

Cílem této práce je analyzovat existující nástroje pro sledování produktů z hlediska poskytovaných služeb, uživatelského hodnocení, podporovaných obchodů a navrhnout a implementovat vlastní řešení tohoto nástroje, který bude podporovat uživatelem registrované e-shopy a implementovat funkce, které pomohou uživateli se sledováním nabídky produktů konkurence a vývoje ceny a dostupnosti zboží napříč registrovanými e-shopy. Posledním úkolem je porovnání vlastní implementace s existujícími řešeními.

Vývoj softwaru probíhá metodou vodopádového modelu. První část práce, která se skládá z druhé až čtvrté kapitoly, se bude zabývat analýzou existujících řešení nástrojů pro sledování produktů jak pro zahraniční tak český trh, které se pokusím popsat z hlediska nabízených služeb, uživatelského hodnocení, oblíbenosti a dostupnosti nástroje, v datové analýze popíšu způsob ukládání dat v relační databázi a ve funkční analýze popíšu funkce aplikace z hlediska důvodu zařazení do mého řešení, výstupy funkcí a jejich případy užití. V páté kapitole se budu zabývat návrhem implementace nástroje, kde popíšu zejména použité technologie a balíčky a v šesté kapitole jeho samotnou implementací. Poslední část bude věnována testování nástroje, kde popíšu výsledky sledování vybraných produktů a navrhnou možná vylepšení a doplňkové funkce nástroje.

## 2 Popis existujících řešení

Tato kapitola se věnuje popisu českých a zahraničních nástrojů pro sledování produktů. Každý z nich je popsán z hlediska dostupných funkcí (popis, využití), e-shopů, pro které je určen, uživatelského hodnocení a dostupnosti (cena, aplikace/rozšíření prohlížeče). K popisu byly vybrány uživateli nejpoužívanější nástroje.

### 2.1 Zahraniční nástroje

Tato podkapitola se věnuje nástrojům určených pro zahraniční trh. Jsou zde popsány zejména řešení určená pro uživatele internetového obchodu Amazon, protože tyto nástroje patří mezi nejpoužívanější a nejoblíbenější na trhu.

#### 2.1.1 Jungle scout

Jungle Scout [1] je nástroj vytvořený za účelem sledování produktů internetového obchodu Amazon. Pro práci s daty využívá databázi různých poboček Amazonu po celém světě. Pomocí nich pak může uživatel filtrovat produkty napříč různými kategoriemi jako jsou například poptávka, cena, hodnocení, prodané kusy aj. Product tracker je funkce aplikace Jungle Scout, která pracuje s real-time daty a informuje uživatele v podobě tabulek a grafů o tom, jak se vybraný produkt snaží prodávat konkurence například formou dárek, akcí a slev. Niche hunter pomáhá uživatelům s novými nápady na produkty, které by mohli na Amazonu prodávat. Pomocí nastavených filtrů a zadaných klíčových slov navrhuje produkty, které jsou pro prodejce výnosné a v jejich prodeji je malá konkurence. Funkce keyword scout pomáhá uživateli s názvem jeho nabídek a se zařazením nabídky do správné kategorie. Tato funkce vygeneruje tisíce klíčových slov, která jsou uživateli použita pro vyhledávání daného produktu na Amazonu a ke každému z těchto slov vypíše počet, kolikrát bylo slovo při vyhledávání použito a kategorii, do které byl produkt zařazen. Navíc ke každému produktu vypisuje průměrnou cenu nabídek. Nástroj Jungle Scout je dostupný jako webová aplikace a jako rozšíření webového prohlížeče Google Chrome. Toto rozšíření poskytuje další výhody při prohlížení produktů na Amazonu. Na základě prohlížené stránky nabízí toto rozšíření odhady prodeje a prognózy zisku. Dále nabízí historickou analýzu produktu formou grafů, které zobrazují počet prodaných kusů za měsíc, cenu produktu a předpoklady prodeje [2]. Ke každému produktu se přiřazuje Opportunity score, které ho ohodnotí podle poptávky, konkurence a uvedených údajů. Díky tomu může uživatel posoudit, zdali je produkt dobrou investicí. Jungle scout je uživateli na stránce Trustpilot ohodnocen nejlepší možnou známkou a bez nadsázky můžeme říct, že se jedná o nejlepší nástroj pro sledování produktů [3]. Webová aplikace je dostupná formou měsíčního předplatného ve třech verzích v rozmezí od 39 do 99 dolarů. Verze se liší v počtu dostupných funkcí, počtu uživatelů a počtem sledovaných produktů. Rozšíření webového prohlížeče Google Chrome je dostupné za jednorázový poplatek

97 dolarů za verzi Lite, která neobsahuje integraci s webovou aplikací a některé funkce. Za 197 dolarů je dostupná verze Pro, která obsahuje veškeré funkce a integraci s webovou aplikací.

### 2.1.2 Unicorn smasher

Tento nástroj je dalším z mnoha, které jsou určeny pro sledování produktů na Amazonu. Jedná se o rozšíření webového prohlížeče Google Chrome [4]. Uživatelům této aplikace pomáhá organizovat jejich prozkoumávání produktů na Amazonu pomocí „nástěnky“, do které si uživatel uloží produkt, který na Amazonu našel a rád by o něm měl informace na jednom místě. Jedná se o hlavní funkci nástroje, ve které uživatel spravuje své sledované produkty, sleduje počet prodávajících daný produkt, jeho cenu, odhadovaný počet prodaných kusů, odhadovaný zisk z prodeje a ohodnocení opportunity score, které slouží jako indikátor, zdali je produkt dobrou investicí. K zjednodušení přístupu si může uživatel své průzkumy označit jako oblíbené a tím mít důležitá data pro svůj obchod vždy a rychle po ruce. Dále nabízí možnost exportu dat do souboru ve formátu csv pro možnost tisku či další zpracování jinou aplikací. Pro sdílení svých průzkumů s dalšími uživateli nástroje se využije funkce diskrétního sdílení. Tato funkce vytvoří obraz pro sdílení, který místo názvu produktu obsahuje jeho kategorii a tím neodhalí ostatním daný produkt, pouze poskytne důležité informace pro poskytnutí zpětné vazby. Unicorn smasher má mezi uživateli průměrné hodnocení, kdy jeho největší výhodou je, že je zadarmo a funguje velmi dobře pro základní průzkum trhu. Jeho nevýhodou je nepřesnost. Jeden z uživatelů uvedl, že jeho produkt, který se neprodával dobře, byl nástrojem ohodnocen jako úspěšný [5].

### 2.1.3 AMZ Tracker

AMZ Tracker [6] je určen pro uživatele, kteří obchodují na trhu Amazonu. Jedná se o webovou aplikaci, která je dostupná v rámci měsíčního předplatného se sedmidenní zkušební dobou. Má čtyři verze v cenovém rozmezí od 50 do 400 dolarů, kdy v každé verzi jsou funkce různě limitovány počtem spravovaných produktů a počtem dostupných výsledků. Tento nástroj dělí své funkce podle tří strategií – ofensivní, defensivní a průzkumná. V ofensivní strategii nabízí funkce Keyword tracking a Super URLs. Keyword tracking poskytuje hodnocení klíčových slov produktů uživatele a jejich postavení v celkovém hodnocení. Dále tato funkce sleduje produkty konkurence, takže uživatel může zjistit, kdy a jakou změnu, která se pozitivně projevila na hodnocení produktu, konkurenční prodejci udělali. Super URLs slouží k vytváření odkazů, které po kliknutí přesměrují uživatele sociálních sítí na produkt na Amazonu. Slouží jako výborná marketingová pomůcka. V defensivní strategii se nachází funkce Negative review alert. Negative review alert zasílá upozornění kdykoliv někdo zanechá recenzi, která byla horší než 5 hvězdiček (1-4 hvězdiček). Díky tomu můžeme rychle reagovat na negativní hodnocení produktu, zjistit jeho nedostatky, chyby při jeho prodeji a tím získat šanci na zlepšení hodnocení. Průzkumná strategie využívá funkcí Keyword research tool a Competitor analysis. Pomocí Keyword research tool najdeme slova a fráze, která uživatelé používají pro vyhledávání na Amazonu a tím nalez-

neme ta správná klíčová slova pro naše produkty. Competitor analysis nám pomůže zjistit chyby a nedostatky v produktech konkurentů. Zobrazuje změny, které provádějí a jak se projevují na celkovém hodnocení produktu. AMZ Tracker je uživateli hodnocen velmi dobře. Jako velké pozitivum uživatelé uvedli přehledné uživatelské rozhraní a kvalitně vytvořená videa pro začátečníky [7]. Mnoho uživatelů se shodlo na tom, že nástroj nenabízí o moc kvalitnější funkce než nástroje, které jsou bezplatné. Další nevýhodou nástroje je špatně fungující zákaznická podpora [8].

#### **2.1.4 Cheap Shark**

Nástroj Cheap shark [9] sleduje ceny počítačových her ze sedmnácti různých obchodů. Jedná se o webovou aplikaci, která je provozována bezplatně a bez nutnosti jakékoliv registrace. U jednotlivých produktů monitoruje jejich cenu, slevy, hodnocení recenzí z různých serverů, datum vydání a skóre nabídky, které je vypočítáno podle velikosti slevy, hodnocení v recenzích a data vydání hry. Uživatel si může na jednotlivé produkty nastavit upozornění, které mu bude posláno na zadanou emailovou adresu, když cena produktu dosáhne cílené částky. Pro vyhledávání produktů slouží filtry, ve kterých můžeme nastavit vyhledávání podle zadané fráze, produkty ve slevě, cenového rozsahu a hodnocení na platformě Steam. Pro každý nastavený filtr můžeme získat RSS feed.

### **2.2 České nástroje**

Následující podkapitola se věnuje nástrojům určených pro český a slovenský trh. Všechny z níže uvedených řešení se věnují všem e-shopům bez ohledu na prodáváný sortiment zboží.

#### **2.2.1 Price-check**

Pricecheck [10] je českým nástrojem pro sledování, monitorování a porovnávání cen produktů na libovolném online trhu (Česko, Slovensko, Německo, Francie, Itálie, Polsko). Webová aplikace je vytvořena pro online obchody, kterým pomáhá s tvorbou té správné ceny nabízených produktů [11]. Druhou skupinou uživatelů jsou dodavatelé a distributoři, kterým tento nástroj pomůže stabilizovat cenu jejich produktů [12]. Hlavní funkcí nástroje je monitoring dat. Monitoring probíhá ve třech automatizovaných krocích. Prvním krokem je nahrání uživatelských dat v libovolném formátu a přes libovolné prostředí, druhým krokem je párování nahraných dat s daty jiných uživatelů včetně správných variant produktů (například velikost, barva) a posledním krokem je vytváření reportů v libovolném formátu, které jdou dělit podle libovolných parametrů. Další funkcí je porovnávání cen, pomocí které zjistíme, jak si náš produkt stojí v porovnání s produkty ze stejného sortimentu, kdo je naší konkurencí a která nabízí levněji. Skladová dostupnost nás informuje o dostupnosti produktu spolu s její historií. Pomocí real-time monitoringu nástroj rychle vyhodnotí, kdo změnil cenu a zašle upozornění s informací o této změně. Z nashromážděných dat nástroj vytváří grafy levnějších cen (kdo prodává levněji, v jakém časovém období), průnik sortimentů (kdo nabízí stejné zboží), pohyby cen (změna cen produktů v rámci dne).

Všechny tyto grafy jsou dostupné na nástěnce, kde kromě těchto grafů najdeme také detailní informace o produktu, možnost procházení dat konkurence [13].

### 2.2.2 Dataweps

Společnost Dataweps rozdělila sledování produktů do pěti různých nástrojů – Azor, Beed, Disivo, SocialWatch a TrendLucid. Každý z těchto nástrojů představuje sadu funkcí, které vykonávají jednu činnost jako například monitoring cen, cenotvorba a monitoring zpětné vazby. Azor [14] je nástroj, který slouží pro monitoring cen na internetu. Azor páruje data podle veškerých informací, pravidelně kontroluje a čistí získané informace a učí se, jak produkty rozpoznat a spárovat. Aplikace neustále prohledává internet a zjišťuje nové informace o produktech. Veškerá získaná data jsou k dispozici ke stáhnutí ve formátech xml, csv a txt pro případné další zpracování. Pro tento monitoring cen stačí jen zadat, jaké zboží na jakých e-shopech chcete sledovat. Beed [15] je nástroj pro automatizované bidování. Podle uživatelem nastavených pravidel automaticky mění cenu za proklik v xml feedu. Beed pro jistotu neustále aktuálního bidu kontroluje pozice až dvanáctkrát denně a pro maximální přesnost počítá výkon produktů podle měření z Google Analytics. Tato funkce funguje na české a slovenské Heuréce a na Zboží.cz. Disivo [16] pomáhá uživatelům s cenotvorbou jejich produktů pomocí cenové analytiky, díky které můžeme zkoumat aktuální situaci na trhu a hledat prostor pro optimalizaci cen. Nástroj zvládne podle nastavených strategií a modelů sám měnit cenu produktu i několikrát denně. Jedna z funkcí nově nastavené strategie zanalyzuje podle dopadu na obrát, zisk, poptávku a uživateli podá hlášení o efektivnosti strategie. SocialWatch [17] slouží pro monitoring zpětné vazby na sociálních sítích (Facebook, Twitter, Instagram) a diskuzních fórech. Nástroj pomáhá s marketingovou analýzou a strategií, informuje o kladných a negativních hodnoceních a internetových kampaních konkurence. Nástroj TrendLucid slouží k zjištění aktuálních trendů, které uživateli v jeho e-shopu chybí a cenovému postavení v jednotlivých kategoriích. TrendLucid [18] poskytuje report, který obsahuje spoustu dat o tom, které produkty lidé nakupují, které přestávají kupovat, kterým e-shopům se momentálně daří, porovnání počtu top produktů v mém e-shopu v porovnání s konkurencí, porovnání průměrných cen v kategoriích. Nástroje od Dataweps jsou hodnoceny velmi dobře a jsou používány předními prodejci v České republice a na Slovensku.

### Shrnutí

Ze získaných informací o výše uvedených nástrojích můžeme prohlásit, že hlavní funkcí těchto aplikací je získávání dat o produktech, zejména jejich cena, dostupnost, počet prodaných kusů a velikost konkurence prodávající daný produkt. Z těchto dat každý nástroj vypočítává odhadovaný obrát, zisk a vytváří grafy vývoje ceny, dostupnosti a zaplnění skladů. Mezi aplikacemi pro Amazon je velmi oblíbená funkce Keyword scout, která slouží pro získání těch nejčastěji používaných klíčových slov pro vyhledávání našeho produktu. Většina aplikací také využívá funkce pro zasílání upozornění či exportu dat do různých formátů.



### 3 Popis zadání

Tato část práce se věnuje obecnému popisu nástrojů pro sledování produktů a popisu vlastního řešení. První podkapitola vysvětlí účel těchto nástrojů, použití, způsob jejich fungování a uvede důvody k jejich využití. Druhá podkapitola bude věnována popisu vlastního řešení nástroje, jeho zaměření na sortiment zboží, zvolené implementační prostředí a použitou databázovou technologii.

#### 3.1 Obecný popis nástrojů

Nástroje pro sledování produktů jsou určeny zejména pro uživatele, kteří vlastní firmu prodávající nějaký sortiment zboží, začínající podnikatele s firmou pro prodej zboží nebo obyčejné lidi, kteří mají prodej produktů jako přivýdělek. Tyto nástroje sledují data o produktech (cena, dostupnost, parametry...) a prodejcích (zaplnění skladů, sortiment zboží...) a uživatelům pomáhají pomocí těchto dat dosáhnout co největších zisků a popularity na trhu. Získané informace prezentují uživatelům formou tabulek a grafů. Data se sbírají různými způsoby například využitím API serverů, přímým přístupem do databáze, kde jsou data uložena nebo parsováním html dokumentů. Jak tyto nástroje dokáží zvýšit zisk? Veškeré nashromážděné informace o produktech jsou zpracovávány různými funkcemi, které daný nástroj obsahuje, data z jednotlivých e-shopů jsou mezi sebou porovnávána a výsledky jsou prezentovány uživateli, který díky tomu dokáže posoudit, zdali produkt prodává draž než konkurence, levněji nebo jestli může cenu zvýšit a pořád zůstane nejlevnějším prodejcem. Takhle, pomocí všech nástrojů pro sledování produktů, dokáže uživatel správně regulovat cenu svých produktů a tím dosahovat co největších zisků.

#### 3.2 Popis vlastního řešení

Pro vývoj vlastního řešení nástroje pro sledování produktů jsem zvolil platformu .NET v jazyce C# z důvodu osobní preference před jinými jazyky a platformami. Pro ukládání dat jsem využil SQL databázi MS SQL, protože integrace MS SQL serveru do aplikace v technologii je jednoduchá a nevyžaduje instalaci různých ovladačů a konektorů. Nástroj je implementován jako webová aplikace, jejíž sortimentem sledovaných produktů jsou sběratelské mince a bankovky prodávané smluvními partnery České národní banky. Uživatelské rozhraní je vytvořeno kaskádovými styly CSS a je přizpůsobeno mobilním zařízením pomocí responzivního designu.

#### Shrnutí

V této kapitole jsme se dozvěděli hlavní funkci všech nástrojů pro sledování produktů, kterou je sběr a prezentace dat o produktech uživatelům, jak tyto informace nástroje získávají a jak z těchto dat dokáží uživatelé zvýšit svůj zisk z prodeje zboží. Dále jsme se blíže seznámili s použitými technologiemi při implementaci vlastního řešení, důvody jejich použití a představili si sortiment zboží, které tento nástroj sleduje.

## 4 Datová a funkční analýza

Tato kapitola se v první části věnuje popisu způsobu ukládání dat v SQL databázi. Jednotlivé atributy v tabulkách jsou popsány z hlediska datového typu, velikosti atributu (zejména u datových typů CHAR a VARCHAR), zdali se jedná o primární nebo cizí klíč, jestli může obsahovat hodnotu NULL a integritního omezení pomocí datového slovníku. Vztahy mezi tabulkami jsou zobrazeny v relačním modelu databáze. U jednotlivých tabulek jsou popsány významy jejich atributů a vztahů s jinými tabulkami. V druhé části je popis jednotlivých funkcí, které tento nástroj obsahuje. Jednotlivý popis obsahuje nastínění toho, jak daná funkce funguje, k čemu slouží, jaké jsou požadované vstupy a co je výstupem funkce.

### 4.1 Datová analýza

V této podkapitole je popsáno schéma relační databáze, určené pro ukládání dat v rámci aplikace, formou datových slovníků a popisem jednotlivých vztahů mezi tabulkami. Na konci podkapitoly je zobrazen relační model databáze.

#### 4.1.1 Datové slovníky

##### Tabulka: uživatel

Do tabulky *uzivatel* se ukládají informace o všech uživateli aplikace. *uID* je identifikační číslo pro uživatele v databázi. Atributy *jmeno* a *prijmeni* slouží, jak už z jejich názvu vyplývá, pro uložení jména a příjmení uživatele. Pro uložení kontaktních údajů se emailová adresa uživatele uchovává v atributu *email*. Tabulka obsahuje dvě integritní omezení, sloupec *email* je omezen regulárním výrazem "%\_@\_%.\_\_%", který zajistí, že před znakem `.` bude alespoň jeden znak, za ním alespoň dva znaky, poté tečka a za ní dva a více znaků. Atribut *password* musí obsahovat 6 a více znaků.

Název	Datový typ	Velikost	Klíč	NULL	IO
uID	Integer	-	PK	NOT NULL	Ne
jmeno	VARCHAR	20	-	NOT NULL	Ne
prijmeni	VARCHAR	25	-	NOT NULL	Ne
email	VARCHAR	50	-	NOT NULL	Ano
password	VARCHAR	254	-	NOT NULL	Ano

Tabulka 1: Datový slovník tabulky *uzivatel*

##### Tabulka: product

Tabulka *product* slouží k ukládání produktů z registrovaných e-shopů. Atribut *pID* je identifikační číslo produktu, které je pro každý záznam unikátní a slouží jako primární klíč k této

tabulce, *eID* je cizí klíč z tabulky *eshop* a určuje, do kterého e-shopu daný produkt patří. *pcID* je druhý cizí klíč a zařazuje produkt do kategorie. Aktuální cena produktu v měně CZK je uložena v *current\_price*, název zboží je ve sloupci *name* a jeho dostupnost je uložena jako bitová hodnota, tedy stav 1 nebo 0, s názvem *available*. Odkaz na internetovou stránku s produktem je uložen v atributu *url*. Aktuálnost ceny a dostupnosti nás informuje datum získání těchto dat ve sloupci, který se nazývá *date*. Tabulka *product* neobsahuje žádná integritní omezení.

Název	Datový typ	Velikost	Klíč	NULL	IO
pID	Integer	-	PK	NOT NULL	Ne
eID	Integer	-	FK	NOT NULL	Ne
pcID	Integer	-	FK	NOT NULL	Ne
name	VARCHAR	60	-	NOT NULL	Ne
current_price	Integer	-	-	NOT NULL	Ne
available	Bit	1	-	NOT NULL	Ne
url	VARCHAR	255	-	NOT NULL	Ne
date	DATE	-	-	NOT NULL	Ne

Tabulka 2: Datový slovník tabulky product

#### Tabulka: eshop

V této tabulce jsou uloženy informace o registrovaných e-shopech. Jako identifikace jednotlivých záznamů v rámci databáze nám k tomuto účelu slouží identifikační číslo *eID*. Název e-shopu, který je získán při registraci ze zadané url pomocí regulárního výrazu, aplikace uloží do atributu *name*. Sloupce *XPath\_name*, *XPath\_price* a *XPath\_items* slouží k ukládání XPath cest, které slouží k parsování html dokumentu a získání dat o produktech z tohoto e-shopu. Na žádný atribut z této tabulky není nastaveno integritní omezení.

Název	Datový typ	Velikost	Klíč	NULL	IO
eID	Integer	-	PK	NOT NULL	Ne
name	VARCHAR	40	-	NOT NULL	Ne
XPath_name	VARCHAR	255	-	NOT NULL	Ne
XPath_price	VARCHAR	255	-	NOT NULL	Ne
XPath_items	VARCHAR	255	-	NOT NULL	Ne

Tabulka 3: Datový slovník tabulky eshop

#### Tabulka: user\_eshop

Tato tabulka slouží k uložení vazby M:N mezi uživatelem a e-shopem. Obsahuje identifikační

číslo záznamu *ueID*, cizí klíče *eID* pro identifikaci e-shopu a *uID* pro identifikaci uživatele v této vazbě. Na žádný z těchto sloupců není aplikováno integritní omezení.

Název	Datový typ	Velikost	Klíč	NULL	IO
ueID	Integer	-	PK	NOT NULL	Ne
uID	Integer	-	FK	NOT NULL	Ne
eID	Integer	-	FK	NOT NULL	Ne

Tabulka 4: Datový slovník tabulky user\_eshop

#### Tabulka: tracked

Další tabulka, která slouží pro uložení vazby M:N, je *tracked*. Slouží k uložení produktů, které uživatel sleduje. Obsahuje tři sloupce, *tID* pro identifikaci záznamů tabulky v databázi, *uID* pro identifikaci uživatele a *pcID* pro specifikaci sledovaného produktu.

Název	Datový typ	Velikost	Klíč	NULL	IO
tID	Integer	-	PK	NOT NULL	Ne
uID	Integer	-	FK	NOT NULL	Ne
pcID	Integer	-	FK	NOT NULL	Ne

Tabulka 5: Datový slovník tabulky tracked

#### Tabulka: notify

Veškerá uživatelem nastavená upozornění jsou uložena v tabulce *notify*. Primárním klíčem této tabulky je číslo *nID*. Pro identifikaci uživatele a produktu, kterého se nastavené upozornění týká, slouží cizí klíče *uID* a *tID* z tabulek *uzivatel* a *tracked*. Zda-li má aplikace poslat uživateli upozornění, když dojde ke změně dat sledovaného produktu, pozná podle hodnoty bitového atributu *onChange*. Sloupec *last\_notify* obsahuje datum naposledy zaslání upozornění. Uživatel si může nastavit pravidelnost zasílání upozornění pomocí atributu *interval*. Ten obsahuje počet dní, po kterém se má zaslat nové hlášení. Na tento sloupec je nastaveno integritní omezení, které říká, že *interval* nesmí být záporné číslo.

Název	Datový typ	Velikost	Klíč	NULL	IO
nID	Integer	-	PK	NOT NULL	Ne
tID	Integer	-	FK	NOT NULL	Ne
uID	Integer	-	FK	NOT NULL	Ne
last_notify	DATE	-	-	NULL	Ne
interval	Integer	-	-	NOT NULL	Ano
onChange	Bit	-	-	NOT NULL	Ne

Tabulka 6: Datový slovník tabulky notify

#### Tabulka: product\_cat

V tabulce *product\_cat* jsou uloženy informace o kategoriích produktů. Obsahuje identifikační číslo záznamů *pcID*. Ve sloupci *category* je obecný název kategorie například "Zlaté mince" a v atributu *name* je uložen název podkategorie například "Zlatá mince Hrad Rabí". Každý záznam v tabulce obsahuje informace o počtu produktů v dané kategorii *number\_of\_products*, počet dostupných produktů *number\_of\_available* a cenové rozpětí dané kategorie pomocí atributů *max\_price* a *min\_price*.

Název	Datový typ	Velikost	Klíč	NULL	IO
pcID	Integer	-	PK	NOT NULL	Ne
category	VARCHAR	100	-	NOT NULL	Ne
name	VARCHAR	100	-	NOT NULL	Ne
number_of_products	Integer	-	-	NOT NULL	Ne
number_of_available	Integer	-	-	NOT NULL	Ne
min_price	Integer	-	-	NOT NULL	Ne
max_price	Integer	-	-	NOT NULL	Ne

Tabulka 7: Datový slovník tabulky product\_cat

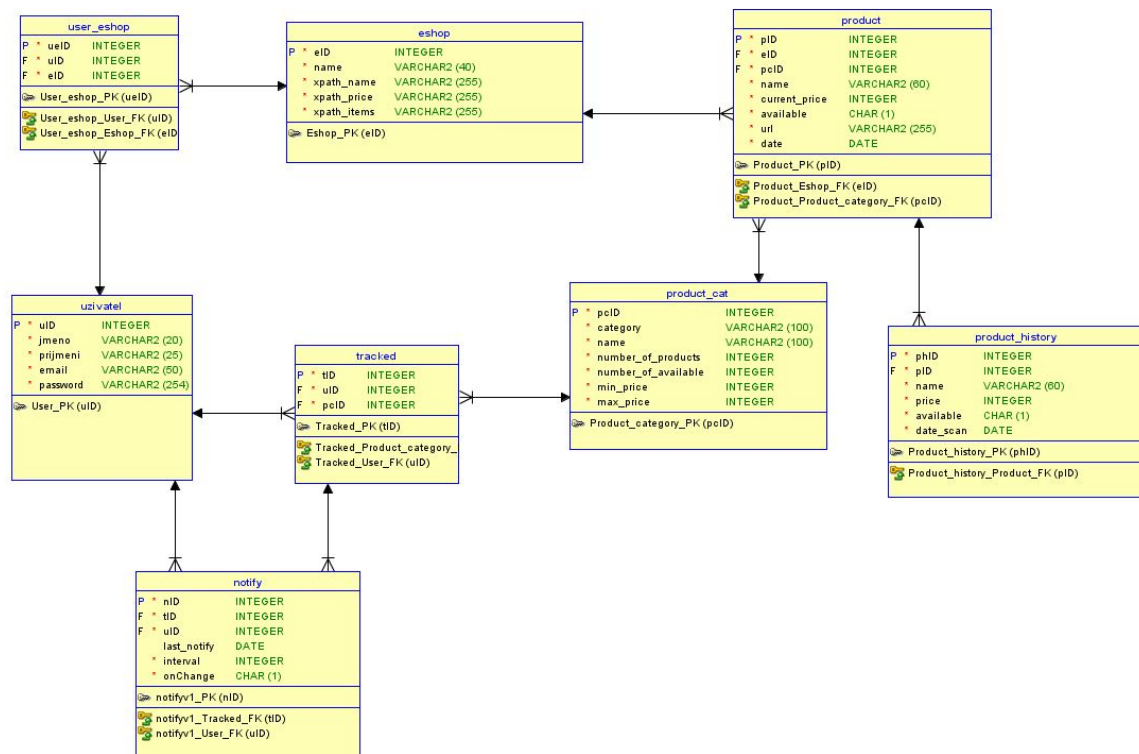
#### Tabulka: product\_history

Historická data produktů se ukládají do tabulky *product\_history*. Primárním klíčem je identifikační číslo *phID*. Pro rozpoznání, kterému produktu daný historický záznam patří, slouží cizí klíč *pID* z tabulky *product*. Jednotlivé řádky obsahují název produktu ve sloupci *name*, cenu produktu a jeho dostupnost v attributech *price* a *available*. K určení, jak stará data daný záznam obsahuje, slouží atribut *date\_scan*. Sloupce v této tabulce nemají nastavená žádná integritní omezení.

Název	Datový typ	Velikost	Klíč	NULL	IO
phID	Integer	-	PK	NOT NULL	Ne
piD	Integer	-	FK	NOT NULL	Ne
name	VARCHAR	60	-	NOT NULL	Ne
price	Integer	-	-	NOT NULL	Ne
available	Bit	1	-	NOT NULL	Ne
date_scan	DATE	-	-	NOT NULL	Ne

Tabulka 8: Datový slovník tabulky product\_history

#### 4.1.2 Relační model



Obrázek 1: Relační model databáze

#### 4.1.3 Vztahy mezi tabulkami

V databázi existují mezi tabulkami vztahy, které definují, kolik záznamů z tabulky A odpovídá záznamům v tabulce B a každý vztah má svůj název a popis, co daný vztah představuje. V databázi této aplikace je těchto vztahů několik. Vztah *Sleduje*(*uzivatel*, *tracked*) představuje uživatele a jeho sledované produkty. Jedná se o vazbu 1:N, kdy jeden uživatel sleduje spoustu produktů, ale jeden záznam v tabulce *tracked* odpovídá jednomu uživateli. Mezi tabulkami *notify* a *tracked*

je vztah *Je reportován(notify, tracked)*, který říká, že na sledovaný produkt je zasíláno hlášení. Tabulka *uzivatel* má vazbu 1:N na tabulku *notify* se vztahem *Má upozornění(uzivatel, notify)* a ten nám říká, že uživatel má nastavený report na sledovaný produkt. Vztahem *Má eshop(uzivatel, user\_eshop)* se rozumí, že uživatel aplikace je majitelem eshopu, nebo ho má pouze jako oblíbený. Pro tabulku *eshop* existují dva vztahy, jedním z nich je *Je v oblíbených(eshop, user\_eshop)*. Ten nám říká, že jeden z uživatelů si přidal tento eshop do oblíbených. Druhým vztahem je *Má produkt(eshop, product)*, který specifikuje, jaké produkty daný eshop prodává. Tabulka *product* má také dva vztahy. Prvním vztahem je *Má historii(product, product\_history)*, který udává historické ceny daného produktu. Jeden produkt má N historických záznamů, jeden řádek v *product\_history* patří jednomu produktu. Druhý vztah je *Má kategorii(product, product\_cat)*. Ten představuje, jaké kategorie je vybraný produkt. Každý produkt má jednu kategorii a v té může být více produktů. Posledním vztahem v databázi je *Je sledován(product\_cat, tracked)*, který říká, že tuto kategorii produktů sleduje některý z uživatelů. Kategorie může být sledována N uživateli, ale jeden záznam v tabulce *tracked* obsahuje jednu kategorii.

## 4.2 Funkční analýza

### Funkce registrace uživatele

Tato funkce slouží k vytvoření nového účtu, díky kterému bude moci uživatel využívat funkce, které jsou přístupné pouze přihlášeným uživatelům. Funkce je implementována jako webový formulář, do kterého osoba, která si chce vytvořit nový účet, zadá své jméno, příjmení, emailovou adresu a heslo. Po odeslání formuláře se zkontroluje, jestli uživatel vyplnil všechna povinná políčka a zdali zadaná emailová adresa a heslo splňují integritní omezení atributů pro uložení těchto dat v databázi. Na konci je uživatel informován formou zprávy vypsané pod registračním formulářem o úspěšné registraci nebo o neúspěšné registraci z důvodu špatně vyplněných údajů.

#### Use case 1

Název: Registrace uživatele

Cíl: Registrovat nového uživatele

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k zadání registračních údajů.
2. Uživatel zadá registrační údaje.
3. Uživatel odešle formulář.
4. Aplikace zkontroluje správnost zadaných údajů.
  - (a) Aplikace zjistí nesprávnost zadaných údajů.

- (b) Aplikace vypíše upozornění.
  - (c) Aplikace vyzve uživatele k opětovnému zadání údajů.
5. Aplikace uloží nového uživatele do databáze.
  6. Aplikace vypíše zprávu o úspěšné registraci.

### **Funkce přihlášení uživatele**

Pomocí této funkce se uživatel přihlásí do aplikace a získá přístup k ostatním funkcím tohoto nástroje a možnost spravovat své sledované produkty a upozornění. Jedná se o webový formulář, do kterého uživatel zadá svou emailovou adresu, kterou použil při registraci účtu, vyplní políčko pro heslo a poté odešle formulář ke zpracování. Aplikace vyhledá v databázi uživatele registrovaného pod zadanou emailovou adresou a zkontroluje, zdali se heslo, které uživatel vyplnil ve formuláři, shoduje s heslem, které bylo uloženo v databázi k dané emailové adrese. Pokud se hesla neshodují, nebo záznam se zadanou emailovou adresou neexistuje, aplikace vypíše upozornění o nesprávně vyplněné emailové adrese nebo heslu. Při nalezené shodě aplikace přihlásí uživatele.

### **Use case 2**

Název: Přihlášení uživatele

Cíl: Přihlásit uživatele do aplikace

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k zadání přihlašovacích údajů.
2. Uživatel zadá údaje.
3. Uživatel odešle formulář.
4. Aplikace zkontroluje správnost zadaných údajů.
  - (a) Aplikace zjistí, že se hesla neshodují nebo emailová adresa v databázi neexistuje.
  - (b) Aplikace vypíše upozornění o špatně zadaných údajích.
  - (c) Aplikace vyzve uživatele k zadání nových údajů.
5. Aplikace přihlásí uživatele.



### **Funkce přidání produktu do sledovaných**

Pokud chce mít uživatel přehled o produktech, které ho zajímají a rád by dostával upozornění, kdykoliv produkt změní svou cenu či dostupnost, musí si ho nejprve přidat do sledovaných. K tomuto účelu slouží tato funkce. Při procházení seznamu produktů má uživatel možnost si vybraný produkt přidat do seznamu sledovaných pomocí kliknutí na tlačítko "sledovat". Funkce poté zkontroluje, jestli už uživatel zvolený produkt sleduje. Pokud ano, vypíše upozornění, že vybraný produkt již sleduje. V opačném případě uloží nový záznam do tabulky *tracked* a vypíše oznámení o přidání produktu na seznam sledovaných.

#### **Use case 3**

Název: Přidání produktu do sledovaných

Cíl: Přidat produkt do seznamu sledovaných.

Účastník: Uživatel, aplikace

1. Aplikace vyčká, než uživatel vybere produkt pro přidání na seznam sledovaných.
2. Uživatel vybere produkt.
3. Aplikace zkontroluje, jestli vybraný produkt uživatel nesleduje.
  - (a) Aplikace zjistí, že uživatel už zvolený produkt sleduje.
  - (b) Aplikace vypíše upozornění, že daný produkt je již uživatelem sledován.
  - (c) Aplikace vyčká na vybrání nového produktu.
4. Aplikace uloží záznam o sledování produktu do databáze.
5. Aplikace vypíše informaci o tom, že byl produkt přidán do seznamu sledovaných.

### **Funkce odebrání produktu ze sledovaných**

Po čase se uživatel může rozhodnout, že některý ze sledovaných produktů ho již nezajímá a nechce ho nadále sledovat. Proto se rozhodne daný produkt odebrat ze svého seznamu sledovaných produktů pomocí této funkce. U jednotlivých produktů v tabulce sledovaných je tato možnost v podobě tlačítka "odebrat". Po kliknutí na ono tlačítko se zobrazí potvrzovací dialog, který se uživatele zeptá, jestli chce vybraný produkt opravdu odebrat ze seznamu sledovaných. Po zvolení možnosti "ano" funkce nejprve odstraní veškerá existující upozornění nastavená na produkt, který je vybrán k odstranění a poté ho odebere ze seznamu sledovaných. Po vykonání této akce, vypíše aplikace informaci o úspěšném odebrání produktu.

#### Use case 4

Název: Odebrání produktu ze sledovaných

Cíl: Odebrat produkt ze seznamu sledovaných

Účastník: Uživatel, aplikace

1. Aplikace vyčká, než uživatel vybere produkt k odebrání se seznamu sledovaných.
2. Uživatel zvolí produkt k odebrání.
3. Aplikace zobrazí potvrzovací dialog.
4. Uživatel zvolí možnost "Ano".
5. Aplikace zkontroluje, jestli jsou na daný produkt nastavená upozornění.
  - (a) Aplikace zjistí, že na produkt jsou nastavená upozornění.
  - (b) Aplikace odstraní upozornění.
6. Aplikace odebere produkt ze seznamu sledovaných.
7. Aplikace vypíše zprávu o úspěšném odebrání produktu ze seznamu.

#### Funkce nastavení upozornění

Jestliže chce uživatel dostávat upozornění formou emailových zpráv kdykoliv, když produkt změni svou cenu či dostupnost, nebo chce pouze pravidelná hlášení o stavu produktu, musí si na sledovaný produkt nastavit upozornění pomocí této funkce. Do webového formuláře zadá uživatel do políčka interval počet dní, jak často chce dostávat hlášení o stavu produktu. Dále si uživatel zvolí, jestli chce dostávat report při každé změně stavu produktu a vybere, ke kterému sledovanému produktu se toto upozornění váže. Po odeslání formuláře se zkontroluje, zdali políčko intervalu neporušuje integritní omezení atributu *interval* v tabulce *notify*. Pokud je vše v pořádku, uloží aplikace nový záznam o upozornění do databáze.

#### Use case 5

Název: Nastavení upozornění

Cíl: Nastavit nové upozornění na sledovaný produkt

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k vyplnění formuláře.
2. Uživatel vyplní formulář.

3. Uživatel odešle formulář.
4. Aplikace zkontroluje správnost zadaných údajů.
  - (a) Aplikace zjistí, že údaje byly zadány špatně.
  - (b) Aplikace vypíše upozornění o špatně zadaných údajích.
  - (c) Aplikace vyzve uživatele k opětovnému zadání.
5. Aplikace uloží záznam o upozornění do databáze.

### **Funkce odebrání upozornění**

Pokud již uživatel nechce dostávat emailové zprávy o stavu produktu, může si svá nastavená upozornění kdykoliv zrušit pomocí funkce pro zrušení upozornění. Ta je dostupná v tabulce s nastavenými upozorněními, která se nachází na stránce "Můj účet". Po kliknutí na tlačítko "odebrat" se zobrazí potvrzovací dialog, který žádá uživatele o potvrzení k vykonání akce. Po zvolení možnosti "Ano" funkce odstraní z databáze záznam o vybraném upozornění a vypíše zprávu o úspěšném provedení akce.

#### **Use case 6**

Název: Odebrání upozornění

Cíl: Odebrat nastavené upozornění na sledovaný produkt

Účastník: Uživatel, aplikace

1. Aplikace vyčká, než uživatel vybere upozornění k odebrání.
2. Uživatel vybere upozornění.
3. Aplikace zobrazí potvrzovací dialog.
4. Uživatel zvolí možnost "Ano".
5. Aplikace odstraní záznam o upozornění z databáze.

### **Funkce přidání e-shopu do oblíbených**

Pomocí této funkce si může uživatel přidat zvolený internetový obchod do oblíbených a tím mít přehled o produktech, které se v tomto obchodě nacházejí a možnost využít další funkci nástroje pro porovnání cen mezi vybranými e-shopy. Pro přidání obchodu do oblíbených si uživatel nejprve vybere ze seznamu registrovaných obchodů ten, který si chce do seznamu přidat. Poté klikne na tlačítko "přidat" a aplikace zkontroluje, jestli se již zvolený e-shop nenachází na seznamu uživatele. Pokud ne, do databáze se uloží nový záznam. Pro uživatele se vypíše informativní zpráva o úspěšném přidání e-shopu na seznam oblíbených.

## Use case 7

Název: Přidání e-shopu do oblíbených

Cíl: Přidat e-shop do seznamu oblíbených

Účastník: Uživatel, aplikace

1. Aplikace vypíše seznam registrovaných e-shopů.
2. Uživatel vybere e-shop k přidání na seznam oblíbených.
3. Uživatel klikne na tlačítko "přidat".
4. Aplikace zkontroluje, jestli nemá uživatel vybraný e-shop ve svém seznamu.
  - (a) Aplikace zjistí, že vybraný e-shop je již v oblíbených.
  - (b) Aplikace vypíše upozornění.
5. Aplikace uloží e-shop do seznamu oblíbených.

## Funkce odebrání e-shopu z oblíbených

Když uživatele přestane e-shop, který má uložený v oblíbených zajímat, může si ho kdykoliv ze svého seznamu odebrat pomocí funkce na odebrání. Odstraněním e-shopu ze seznamu oblíbených ztratí uživatel možnost porovnat ceny produktů z odstraněného internetového obchodu s jiným obchodem. Funkce je dostupná v tabulce, která se nachází na stránce "Moje e-shopy". Ta obsahuje správu oblíbených e-shopů. Uživatel si vybere obchod, který chce ze seznamu odebrat a klikne na tlačítko "odebrat". Zobrazí se potvrzovací dialog, který se uživatele dotáže, jestli si přeje internetový obchod opravdu odebrat. Po zvolení možnosti "Ano" aplikace tento e-shop z uživatelského seznamu odstraní.

## Use case 8

Název: Odebrání e-shopu ze seznamu oblíbených

Cíl: Odebrat e-shop ze seznamu oblíbených

Účastník: Uživatel, aplikace

1. Aplikace vyčká, než uživatel vybere e-shop k odebrání.
2. Uživatel vybere e-shop.
3. Aplikace zobrazí potvrzovací dialog.
4. Uživatel potvrdí odebrání e-shopu ze seznamu oblíbených.
5. Aplikace odebere e-shop.

### **Funkce porovnání cen v e-shopech**

Tato funkce slouží k porovnání cen stejných produktů, které prodávají různé e-shopy. Díky této funkci dokáže uživatel jednoznačně určit, který internetový obchod prodává jaký produkt o kolik procent levněji nebo draž. Pro porovnání cen uživatel vybere e-shop ze svého seznamu oblíbených a jeden z registrovaných e-shopů. Funkce porovná ceny stejných produktů a do tabulky vypíše cenu produktu v jeho obchodě, cenu v porovnávaném e-shopu a procentuální rozdíl cen.

#### **Use case 9**

Název: Porovnání cen v e-shopech

Cíl: Porovnat ceny produktů mezi dvěma e-shopy

Účastník: Uživatel, aplikace

1. Aplikace vyčká, než uživatel vybere e-shopy k porovnání.
2. Uživatel vybere e-shopy.
3. Aplikace porovná ceny a vypočítá procentuální rozdíl.
4. Aplikace vypíše výsledky do tabulky.

### **Funkce přidání produktu**

Pokud chce uživatel přidat do databáze produktů nový záznam a začít sledovat růst a pokles ceny produktu a jeho dostupnost, učiní tak pomocí této funkce. Uživatel zadá URL adresu odkazující na vybraný produkt k registraci a ze seznamu registrovaných kategorií vybere tu, do které produkt spadá. Pokud se požadovaná kategorie v seznamu nenachází, může novou kategorii vytvořit pomocí další funkce nástroje pro přidání nové kategorie. Po zadání těchto dvou vstupů se zkontroluje, zdali je produkt z registrovaného e-shopu. Pokud není, aplikace vypíše hlášení o neznámém obchodu a uživateli doporučí jeho registraci. Pokud je produkt ze známého e-shopu, aplikace pomocí parsování html dokumentu získá údaje o ceně, názvu a dostupnosti produktu a tyto údaje uloží do databáze. Nakonec se kategorie, do které byl produkt zařazen, aktualizuje pomocí uložené procedury *UpdateCatStats*. Uživatel poté dostane hlášení v podobě vypsané hlášky o úspěšném přidání produktu.

#### **Use case 10**

Název: Přidání produktu

Cíl: Přidat nový produkt do databáze

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k zadání URL adresy a vybrání kategorie.

2. Uživatel zadá URL adresu.
  - (a) Uživatel zjistí, že požadovaná kategorie neexistuje.
  - (b) Uživatel zaregistruje novou kategorii. (Use case: Přidání kategorie)
3. Uživatel vybere kategorii.
4. Aplikace zkontroluje, jestli je e-shop registrován.
  - (a) Aplikace zjistí, že se jedná o neznámý e-shop.
  - (b) Aplikace vypíše hlášení, že se jedná o neznámý e-shop.
5. Aplikace získá data o produktu.
6. Aplikace uloží nový produkt do databáze.
7. Aplikace aktualizuje atributy v tabulce *product\_cat*.
8. Aplikace vypíše hlášení o úspěšném přidání produktu.

### **Funkce přidání kategorie**

Každý produkt spadá do nějaké kategorie, která se skládá z obecného názvu kategorie (například "Zlatá mince") a obecného názvu produktu (například "Mince Hrad Rabí"). Tato kategorie se uvádí při přidání nového produktu. Když požadovaná kategorie neexistuje, musí ji uživatel přidat. K tomu využije tuto funkci. Do webového formuláře uživatel zadá obecný název kategorie, obecný název produktu a formulář odešle. Aplikace poté novou kategorii přidá do databáze a uživateli vypíše hlášení.

### **Use case 11**

Název: Přidání kategorie

Cíl: Přidat do databáze novou kategorii

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k zadání názvů kategorie a produktu.
2. Uživatel vyplní údaje.
3. Uživatel odešle formulář.
4. Aplikace zkontroluje, jestli jsou všechna políčka vyplněná.
  - (a) Aplikace zjistí, že některá políčka nejsou vyplněná.
  - (b) Aplikace vypíše hlášení o špatně vyplněném formuláři.

- (c) Aplikace vyzve uživatele k opětovnému zadání.
- 5. Aplikace uloží novou kategorii do databáze.
- 6. Aplikace vypíše hlášení o úspěšné registraci nové kategorie.

### **Funkce registrace e-shopu**

Aby aplikace věděla, jak má parsovat jednotlivé webové stránky e-shopů k získání dat o produktech, musí být internetový obchod registrován. Pro parsování HTML dokumentu potřebujeme XPath cesty, které získáme během procesu registrace a ty uložíme do databáze. Funkce po uživateli požaduje zadání URL adresy libovolného produktu z e-shopu, který chceme registrovat. Poté se uživateli vypíší jednotlivé textové řetězce v dokumentu, ze kterých pomocí checkboxů uživatel vybere ty, které obsahují požadovaná data o produktu. Po vybrání řetězců obsahující název, cenu a dostupnost produktu uživatel odešle formulář. Ze zadané URL adresy aplikace pomocí regulárních výrazů získá název e-shopu a z textových řetězců získá XPath cesty. Tyto informace poté uloží do databáze a pro uživatele vypíše hlášení o úspěšné registraci obchodu.

### **Use case 12**

Název: Registrace e-shopu

Cíl: Registrovat nová e-shop

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k zadání URL adresy.
2. Uživatel zadá URL adresu.
3. Aplikace vypíše textové řetězce v dokumentu.
4. Uživatel vybere textové řetězce k registraci.
5. Uživatel odešle formulář.
6. Aplikace získá požadovaná data z informací ve formuláři.
7. Aplikace uloží e-shop do databáze.
8. Aplikace vypíše hlášení o registraci e-shopu.

### Funkce aktualizace produktu

Když si uživatel myslí, že data o produktu nejsou aktuální, může informace o produktu nechat aktualizovat jednou denně za pomoci této funkce. Možnost aktualizace nalezne uživatel v tabulce s produkty. Po kliknutí na tlačítko "aktualizovat" se zkontroluje, kdy naposledy byla data aktualizována. Pokud se zjistí, že k aktualizaci došlo naposledy před méně než jedním dnem, vypíše se upozornění, že funkce pro aktualizaci dat není pro tento produkt ještě dostupná. Při možnosti aktualizace se z databáze načtou XPath cesty pro parsování e-shopu, ze kterého je aktualizovaný produkt a načtou se nová data. Jestli získaná data značí, že produkt je nedostupný (za pomoci parsování se nepodařilo získat data o názvu produktu nebo ceně), data z tabulky *product* se pomocí uložené procedury *CopyProductToHistory* přepírují do tabulky *product\_history*. Jestliže načtená data byla v pořádku, do databáze uložíme historii produktu a aktualizujeme data o produktu. Nakonec se pomocí uložené procedury *UpdateCatStats* aktualizují atributy *number\_of\_products*, *number\_of\_available*, *min\_price*, *max\_price* v tabulce *product\_history* a všem uživatelům, kteří měli na aktualizovaný produkt nastavené upozornění, se odešle informativní email. O úspěšné aktualizaci produktu je uživatel informován vypsáním hlášením.

### Use case 13

Název: Aktualizace produktu

Cíl: Aktualizovat data o produktech

Účastník: Uživatel, aplikace

1. Uživatel klikne na aktualizaci produktu.
2. Aplikace zkontroluje, jestli je aktualizace možná.
  - (a) Aplikace zjistí, že aktualizace produktu je nedostupná.
  - (b) Aplikace vypíše hlášení o nedostupnosti aktualizace.
3. Aplikace stáhne nová data o produktu.
4. Aplikace zkontroluje stažená data.
  - (a) Aplikace zjistí, že data nejsou v pořádku.
  - (b) Aplikace zkopíruje data do tabulky *product\_history*.
  - (c) Aplikace vypíše hlášení o aktualizaci produktu.
5. Aplikace uloží nová data o historii produktu do databáze.
6. Aplikace aktualizuje data o produktu v databázi.



7. Aplikace odešle email všem uživatelům, kteří mají na tento produkt nastavené upozornění, pokud došlo ke změně hodnoty ceny nebo dostupnosti.
8. Aplikace aktualizuje data v tabulce *product\_cat*.
9. Aplikace vypíše hlášení o úspěšné aktualizaci produktu.

### **Funkce výpisu produktů z internetových srovnávačů**

Tato funkce slouží pro získání seznamu produktů, které jsou v nabídce na stránkách heureka.cz a bazos.cz a díky tomu získá uživatel rychlý přehled o tom, kdo všechno vybraný produkt prodává a za jakou cenu. Uživatel do webového formuláře zadá název hledaného produktu a vybere, kde chce daný produkt hledat. Funkce funguje opět na principu parsování HTML dokumentu, ze kterého získá všechny důležité informace. Získaná data prezentuje formou tabulky, která obsahuje název produktu, jeho cenu a název e-shopu. Pokud hledaný výraz nenalezne žádný výsledek, funkce vypíše pro uživatele hlášení.

#### **Use case 14**

Název: Výpis produktů z internetových srovnávačů

Cíl: Vypsát informace o produktech z internetových srovnávačů

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k zadání hledaného výrazu.
2. Uživatel zadá výraz.
3. Uživatel zvolí srovnávač.
4. Aplikace stáhne výsledky hledaného výrazu.
  - (a) Hledaný výraz nevrátil žádné výsledky.
  - (b) Aplikace vypíše upozornění.
5. Aplikace vypíše výsledky hledání formou tabulky.

### **Funkce XML feedu**

Obchody, které mají na svých stránkách dostupný XML soubor obsahující seznam produktů, které obchod prodává, mohou využít této funkce pro procházení těchto produktů a pomocí ostatních funkcí tyto produkty sledovat nebo je uložit do databáze. Do webového formuláře uživatel vloží URL adresu, na které se XML soubor nachází. Aplikace poté XML soubor z této stránky stáhne a zobrazí ho uživateli ve stránkované podobě, kdy na jedné stránce se nachází 5 záznamů. U každého záznamu může uživatel zvolit "sledovat produkt" nebo "přidat produkt". Pokud uživatel zvolí možnost sledování produktu, tak proběhne kontrola, jestli je vybraný produkt

registrován v databázi. Jestli je produkt registrován, přidá ho do uživatelova seznamu sledovaných produktů. V opačném případě vypíše uživateli hlášku, že se jedná o neznámý produkt. Když uživatel zvolí možnost přidání produktu, tak aplikace na nové záložce zobrazí registraci produktu a do políčka s URL adresou zkopíruje z XML feedu obsah elementu URL obsahující webový odkaz na produkt.

### Use case 15

Název: XML feed

Cíl: Vypsat produkty, které se nachází v XML feedu e-shopu

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k zadání URL adresy odkazující na XML soubor.
2. Uživatel zadá URL adresu.
3. Aplikace zobrazí prvních 5 záznamů v XML souboru.
4. Aplikace čeká na volbu uživatele.
  - (a) Uživatel zvolí načtení další stránky.
    - i. Aplikace zkontroluje, jestli může načíst další stránku.
    - ii. Aplikace zobrazí následujících 5 záznamů.
  - (b) Uživatel zvolí načtení předchozí stránky.
    - i. Aplikace zkontroluje, jestli může načíst předchozí stránku.
    - ii. Aplikace zobrazí předchozích 5 záznamů.
  - (c) Uživatel zvolí přidání produktu do seznamu sledovaných.
    - i. Aplikace zkontroluje, jestli je zvolený produkt registrován.
    - ii. Aplikace přidá produkt do seznamu sledovaných. (Use case: Přidat produkt do sledovaných)
  - (d) Uživatel zvolí přidání produktu do databáze.
    - i. Aplikace zkopíruje obsah elementu, obsahující URL adresu produktu.
    - ii. Aplikace otevře novou záložku s registrací produktu.
    - iii. Aplikace přidá produkt. (Use case: Přidání produktu)

### **Funkce hlídacího psa**

Všem uživatelům jde zejména o to, mít co největší přehled o cenách konkurence a o cenách malých obchodníků, u kterých se naskýtá možnost odkoupení produktu s velkou poptávkou za nízkou cenu a prodání tohoto produktu za daleko vyšší cenu s velkým ziskem. Pro tento účel slouží funkce hlídacího psa, který v pravidelných intervalech skenuje internetový srovnávač heureka.cz a aukční síň bazos.cz a z nich stahuje data o produktech v e-shopu uživatele. Pokud některý z internetových obchodů prodává produkt levněji než obchod uživatele, tak se uživateli zašle emailová zpráva s hlášením o tom, které e-shopy prodávají jaký produkt a o kolik levněji s výpisem všech aukčních nabídek daného produktu na bazos.cz. Funkce je implementována jako Windows služba, která obsahuje časovač na pět hodin. Po uplynutí pěti hodin se spustí funkce pro skenování a zaslání výsledků uživatelům.

#### **Use case 16**

Název: Hlídací pes

Cíl: Informovat uživatele o cenách produktů v pravidelných intervalech

Účastník: Windows služba

1. Služba počká na uplynutí časovače.
2. Služba stáhne data o produktech.
3. Služba načte z databáze data o produktech uživatele.
4. Služba porovná ceny produktů.
5. Služba zašle emailové zprávy s hlášením.

### **Funkce výpisu produktů**

Pro procházení všech produktů, které jsou uloženy v databázi, slouží tato funkce. Uživatel si nejprve vybere, kterou kategorii produktů chce zobrazit (například "Zlaté mince"). Poté aplikace vypíše všechny podkategorie. Zde se nachází možnost sledování produktu, aktualizace dat produktů a výpis všech produktů, které se ve vybrané podkategorii nachází. Při zvolení možnosti výpisu produktů se zobrazí položky z jednotlivých e-shopů z možností výpisu historických cen a grafu vývoje ceny.

#### **Use case 17**

Název: Výpis produktů

Cíl: Vypsát seznam produktů uložených v databázi podle vybraných kategorií

Účastník: Uživatel, aplikace

1. Aplikace vyzve uživatele k vybrání kategorie.
2. Uživatel vybere kategorii.
3. Aplikace zobrazí v tabulce jednotlivé podkategorie.
4. Uživatel zvolí výpis produktů v podkategorii.
  - (a) Uživatel zvolil "sledovat produkt".
    - i. Aplikace přidá produkt do sledovaných. (Use case: Přidat produkt do sledovaných)
  - (b) Uživatel zvolil "aktualizovat produkt".
    - i. Aplikace aktualizuje data produktů. (Use case: Aktualizace produktu)
5. Aplikace vypíše produkty v podkategorii.
6. Uživatel zvolí detailní výpis produktu.
7. Aplikace zobrazí historické ceny produktu.

### **Funkce zasílání pravidelných upozornění**

Tato funkce slouží k zasílání emailových zpráv s pravidelným hlášením o stavu ceny a dostupnosti sledovaných produktů, na kterých má uživatel nastavené upozornění s intervalem zasílání větší než nula. Funkce je implementována jako Windows služba, která každých 24 hodin kontroluje veškerá uživateli nastavená upozornění. Pokud zjistí, že je třeba zaslat uživateli hlášení, tak si z databáze načte data o sledovaných produktech a ty zašle uživateli v emailové zprávě.

#### **Use case 18**

Název: Zaslání pravidelných upozornění

Cíl: Zaslat uživateli hlášení o stavu produktu.

Účastník: Windows služba

1. Služba počká na uplynutí časovače.
2. Služba načte veškerá upozornění uživatele.
3. Služba zkontroluje, jestli se má zaslat upozornění.
4. Služba načte data o sledovaných produktech.
5. Služba zašle uživateli hlášení o jeho sledovaných produktech.

### **Funkce správy účtu**

Pokud chce uživatel spravovat svá nastavená upozornění a seznam sledovaných produktů, dostane se k nim přes správu svého uživatelského účtu. Tato funkce zobrazí uživateli seznam jeho sledovaných produktů a umožní mu jejich odebrání. Dále uživateli vypíše jeho nastavená upozornění a umožní jejich přidání a odebrání.

#### **Use case 19**

Název: Správa účtu

Cíl: Možnost spravovat seznam sledovaných produktů a nastavené upozornění.

Účastník: Uživatel, aplikace

1. Aplikace zobrazí seznam sledovaných produktů a nastavených upozornění.
2. Uživatel zvolí jednu z dostupných možností.
  - (a) Uživatel zvolí odebrání produktu ze sledovaných. (Use case: Odebrání produktu ze sledovaných)
  - (b) Uživatel zvolí nastavení nového upozornění. (Use case: Nastavení upozornění)
  - (c) Uživatel zvolí odebrání upozornění. (Use case: Odebrání upozornění)

### **Funkce správy e-shopů**

Pomocí této funkce uživatel může spravovat své internetové obchody, které chce mít ve svém seznamu oblíbených, nebo chce nějaký obchod z tohoto seznamu odebrat. Pokud chce uživatel přidat e-shop do svého seznamu, který ještě nebyl registrován, může se nechat přesměrovat na registraci nového e-shopu a přidat ho do databáze.

#### **Use case 20**

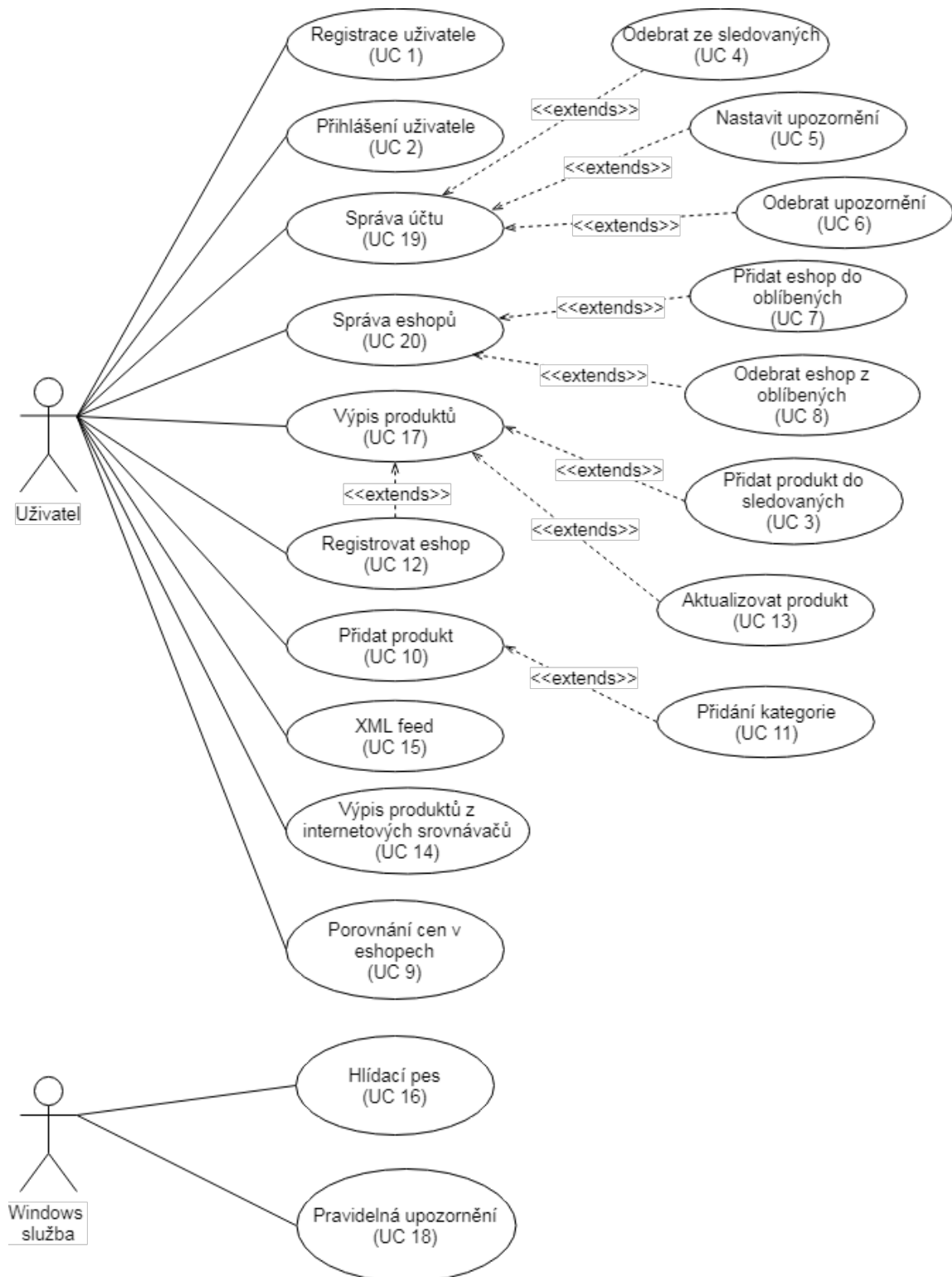
Název: Správa e-shopů

Cíl: Možnost spravovat seznam oblíbených e-shopů.

Účastník: Uživatel, aplikace

1. Aplikace vypíše seznam oblíbených e-shopů.
2. Uživatel zvolí jednu z dostupných možností.
  - (a) Uživatel zvolí přidat e-shop na seznam. (Use case: Přidání e-shopu do oblíbených)
  - (b) Uživatel zvolí odebrat e-shop ze seznamu. (Use case: Odebrání e-shopu z oblíbených)
  - (c) Uživatel zvolí registraci nového e-shopu. (Use case: Registrace e-shopu)

### 4.3 Use case diagram



Obrázek 2: Use case diagram

## Shrnutí

V první části této kapitoly jsme se dozvěděli, jak jsou data uložena v relační databázi MS SQL, podívali jsme se na relační model databáze a přečetli si význam jednotlivých tabulek a jejich atributů. Vysvětlili jsme si význam jednotlivých vztahů mezi tabulkami, co znamenají, proč jsou v databázovém modelu důležité a jakou mají kardinalitu. Pro popis atributů tabulek byl použit datový slovník, který popsal datový typ atributu, velikost, roli primárního klíče, nulovou hodnotu a integritní omezení. Druhá část kapitoly byla věnována funkční analýze nástroje. Jednotlivé funkce byly popsány z hlediska účelu funkce, požadovaných vstupů od uživatele, výstupů funkce a letmého nástinu fungování. Ke každé funkci je popsán případ užití podle šablony název případu užití, jeho cíl, aktéři a hlavní scénář s alternativním průběhem. Na konci kapitoly jsou tyto případy užití poskládány do UML diagramu.

## 5 Návrh implementace

Tato kapitola bude věnována popisu použitých technologií pro implementaci jednotlivých funkcí nástroje a ukládání dat. Kromě technologií budou zmíněny také jednotlivé balíčky tříd a funkcí, které byly důležité pro implementaci. V popisu se zaměřím zejména na důvod jejich využití, výhody, nevýhody a případně důvod upřednostnění vybrané technologie před jinou. Druhá část kapitoly bude věnována návrhu grafického uživatelského rozhraní webové aplikace.

### 5.1 Technologie pro implementaci

Pro vývoj webové aplikace jsem zvolil platformu ASP.NET. Jedním z hlavních důvodů využití této platformy bylo to, že pro vývoj aplikací využívá softwarovou architekturu MVC, která odděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do nezávislých komponent, takže modifikace některé z nich má minimální vliv na ostatní komponenty. Tím je aplikace flexibilnější, snadnější pro vývoj, změna jednotlivých komponent nemá vliv na ostatní části aplikace a lépe se udržuje. Alternativou k MVC by bylo například Autonomous view, které je ale vhodnější pro menší aplikace. Dalším důvodem využití ASP.NET je možnost vytvářet dynamické webové stránky pomocí tzv. Razor engine, což je kód v jazyce C#. Kromě kódu v jazyce C# jde při tvorbě dynamických stránek využít jazyka HTML, JavaScript a CSS. Z důvodu využití platformy .NET pro vývoj aplikace jsem se rozhodl pro ukládání dat vybrat relační databázi MS SQL. Důvodem tohoto rozhodnutí je výborná kompatibilita mezi aplikací na platformě .NET a databází MS SQL, jelikož tato platforma obsahuje třídy pro jednoduché připojení k databázi a pro přenos dat bez nutnosti instalace jakýchkoliv externích balíčků a ovladačů. Tím je zajištěná jednoduchá manipulace s daty mezi databází a aplikací. Pro implementaci grafického uživatelského rozhraní byly použity kaskádové styly CSS. Ty byly navrženy tak, aby splňovaly podmínky responzivního návrhu a bylo možné aplikaci používat i na mobilních zařízeních.

### 5.2 Použité balíčky a knihovny

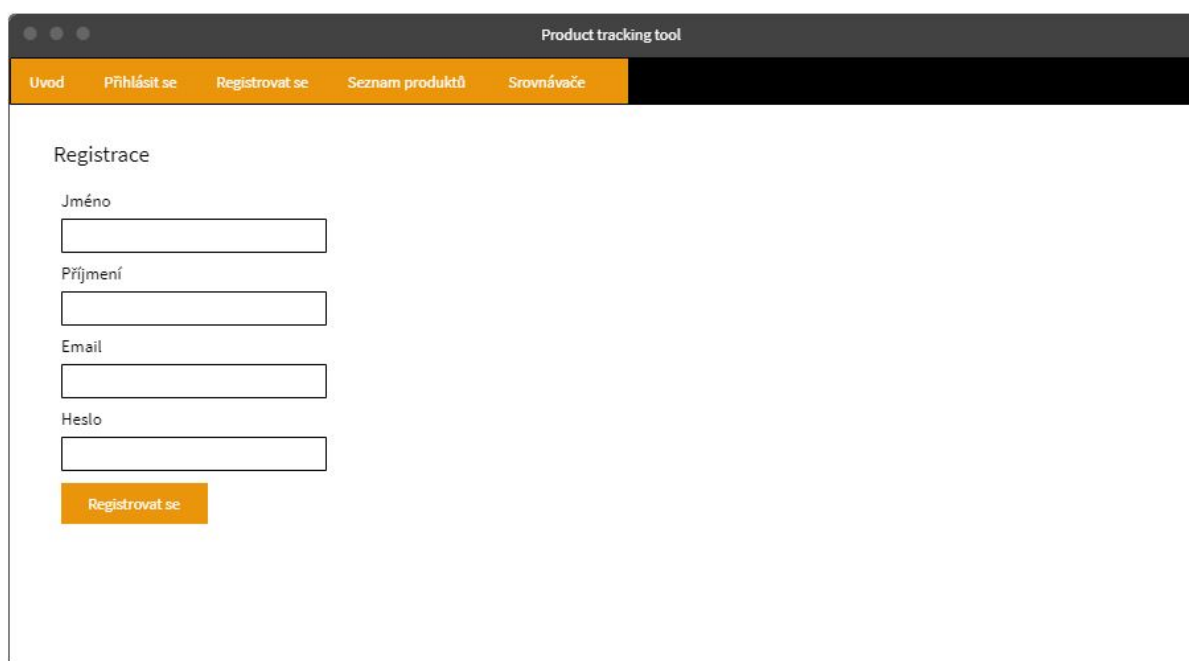
Pro parsování HTML dokumentů je použit balíček s názvem *HtmlAgilityPack*, který je volně k dispozici z distribuce NuGet na platformě .NET. Tento balíček obsahuje třídy, pomocí kterých lze parsovat dokumenty HTML stejným způsobem, jako by se jednalo o dokumenty ve značkovacím jazyce XML. Z tohoto důvodu jsem pro parsování upřednostnil tento balíček před ostatními. Tento balíček obsahuje třídy *HtmlDocument* pro uložení obsahu dokumentu, *HtmlNode* pro uložení jednoho uzlu dokumentu nebo například *HtmlNodeCollection* pro uložení kolekce vybraných uzlů z HTML dokumentu. Pro vybrání požadovaných uzlů se používá dotazovací jazyk XPath, který pomocí zadaného výrazu ve formě cesty, vybere uzel či kolekci uzlů. Alternativami pro parsování HTML na platformě .NET je například NuGet balíček *HtmlMonkey*. Pro přihlášení uživatele do aplikace se používá balíček *Session*. Tento balíček podporuje práci s daty, které jsou uloženy na serveru a jsou svázány s uživatelem pomocí cookies, které obsahují klíč k těmto



datům. Jelikož jsou tyto data uložena na serveru, tak je uživatel nemůže číst ani modifikovat. Jako alternativa pro přihlašování uživatelů do aplikace se dá využít předpřipraveného nástroje pro autentizaci a autorizaci na platformě .NET *Identity*. Grafy jsou v aplikaci vykreslovány pomocí JavaScript knihovny *Chart.js*, která nabízí širokou škálu různých typů grafů. Možnou alternativou pro vykreslování grafů je například další JavaScript knihovna *CanvasJS*.

### 5.3 Uživatelské rozhraní

V této části kapitoly budou zobrazeny nákresy návrhu uživatelského rozhraní webové aplikace. Vzhled jednotlivých webových stránek je vymodelován pomocí nástroje pro tvorbu "wireframů" tak, aby co nejpřesněji odpovídal skutečné podobě aplikace. Vzhled webových stránek je vytvořen pomocí kaskádových stylů CSS.



The image shows a web browser window with the title "Product tracking tool". The navigation bar is orange and contains the following links: "Úvod", "Přihlásit se", "Registrovat se", "Seznam produktů", and "Srovnávače". The main content area is white and contains a registration form titled "Registrace". The form has four input fields: "Jméno", "Příjmení", "Email", and "Heslo". Below the "Heslo" field is an orange button labeled "Registrovat se".

Obrázek 3: Formulář pro registraci uživatele

Na obrázku číslo 3 je formulář pro registraci uživatele, který navštíví nový uživatel pokud se chce do aplikace registrovat a využívat jejích služeb. Uživatel zadá údaje požadované pro registraci a klikne na tlačítko registrovat. Tím je vytvořen pro uživatele nový účet. Po registraci se může uživatel do aplikace přihlásit pomocí formuláře, který je zobrazen na obrázku číslo 4. Do tohoto formuláře uživatel zadá přihlašovací údaje, tedy emailovou adresu a heslo a klikne na tlačítko přihlásit se.

Product tracking tool

Uvod Přihlásit se Registrovat se Seznam produktů Srovnávače

Přihlásit se

Email

Heslo

Přihlásit se

Obrázek 4: Formulář pro přihlášení uživatele

Po přihlášení do aplikace je uživatel přesměrován na stránku "Můj účet", která je zobrazena na obrázku číslo 5. Na této stránce se nachází správa produktů uložených v seznamu sledovaných a správa nastavených upozornění. Po přihlášení se uživateli také zpřístupní další funkce nástroje, které může najít v menu aplikace.

Product tracking tool

Uvod Můj účet Registrovat eshop Přidat produkt Moje eshopy Porovnat ceny XML feed Seznam produktů Srovnávače Odhlásit

Můj účet

Sledované produkty

Název	Odebrat	Produkty
Hrad Zvíkov	Odebrat	Produkty

Upozornění

Nemáte nastavená žádná upozornění

Obrázek 5: Stránka Můj účet

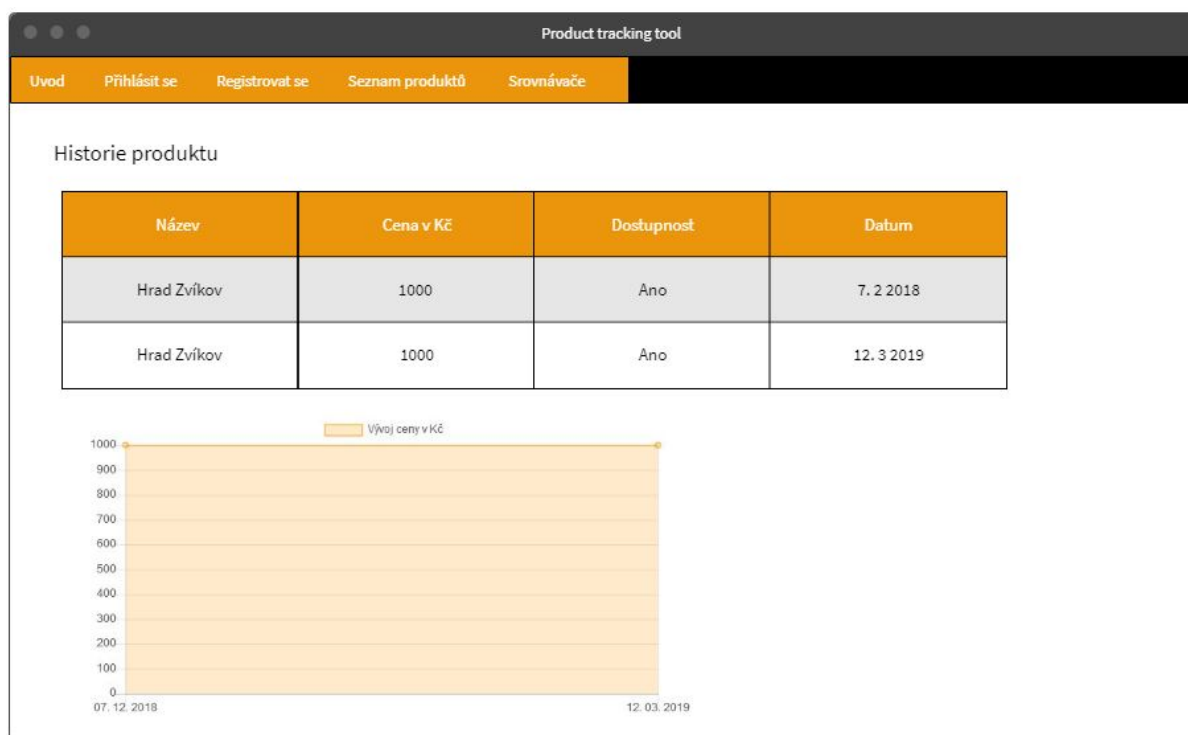
Po kliknutí na záložku "Seznam produktů" v menu aplikace se uživateli nejprve zobrazí seznam kategorií produktů. Uživatel si zvolí, kterou kategorii chce zobrazit a aplikace vypíše seznam

produktů ve zvolené kategorii. U každého produktu má uživatel možnost přidat produkt do seznamu sledovaných, aktualizovat produkty nebo vypsat seznam produktů.

Product tracking tool					
Uvod	Přihlásit se	Registrovat se	Seznam produktů	Srovnávače	
Seznam produktů v kategorii Zlaté mince					
Název	Cenové rozpětí v Kč	Dostupných	Produkty	Sledovat	Aktualizovat
Hrad Zvíkov	17500-16100	3	Produkty	Sledovat	Aktualizovat
Hrad Veveří	5000-1000	0	Produkty	Sledovat	Aktualizovat
Hrad Kost	22900-18300	5	Produkty	Sledovat	Aktualizovat

Obrázek 6: Seznam produktů

Ve výpisu seznamu produktů má uživatel možnost si u každého produktu nechat zobrazit jeho historické ceny a dostupnost. U každého záznamu je přiřazeno datum, kdy měl produkt tento stav. Z těchto historických dat o produktu je vytvořen graf vývoje ceny.



Obrázek 7: Historie produktu

Uživatel může přidat nový produkt do databáze pomocí formuláře "Přidat produkt". V něm uživatel zadá URL adresu produktu a vybere ze seznamu kategorií, do které produkt spadá. Pokud požadovaná kategorie neexistuje, může ji uživatel přidat pomocí možnosti "Přidat kategorii".

The screenshot shows a web browser window titled "Product tracking tool". The navigation bar at the top contains the following links: "Úvod", "Můj účet", "Registrovat eshop", "Přidat produkt", "Moje eshopy", "Porovnat ceny", "XML feed", "Seznam produktů", "Srovnávače", and "Odhlásit". The main content area is titled "Přidat produkt". It contains a text input field labeled "URL". Below the input field are two orange buttons: "Přidat kategorii" and "Přidat".

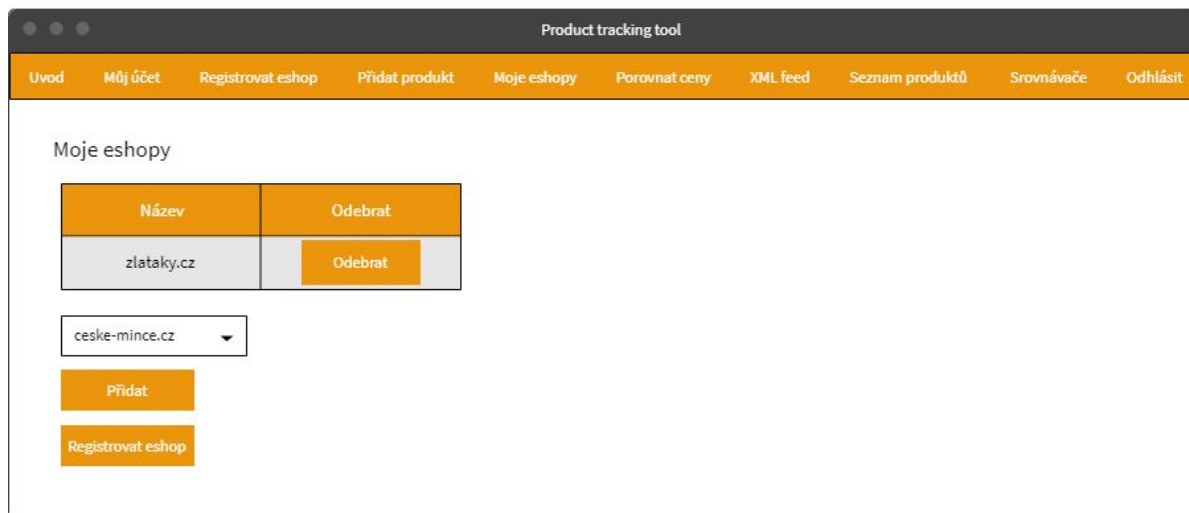
Obrázek 8: Formulář přidání produktu

Pro přidání nového produktu do databáze je kromě vybrání kategorie produktu také potřeba registrovat e-shop, který vybraný produkt prodává. Toho uživatel dosáhne registračním formulářem "Registrovat e-shop". V něm uživatel zadá URL adresu k libovolnému produktu a z vypsaného HTML dokumentu vybere pomocí "checkboxů" elementy, které obsahují data o ceně, názvu a dostupnosti produktu.

The screenshot shows a web browser window titled "Product tracking tool". The navigation bar at the top contains the following links: "Úvod", "Můj účet", "Registrovat eshop", "Přidat produkt", "Moje eshopy", "Porovnat ceny", "XML feed", "Seznam produktů", "Srovnávače", and "Odhlásit". The main content area is titled "Registrovat eshop". It contains a text input field labeled "URL" with the value "https://www.ceske-mince.cz/coin\_gold". Below the input field are two orange buttons: "Zobrazit" and "Registrovat". Below these buttons are three checkboxes: "Název" (checked), "Cena" (unchecked), and "Dostupnost" (unchecked). Below the checkboxes are three lines of HTML code: "<h1>Mince Hrad Veveří </h1>", "<span>16 300 Kč </span>", and a single dot ".".

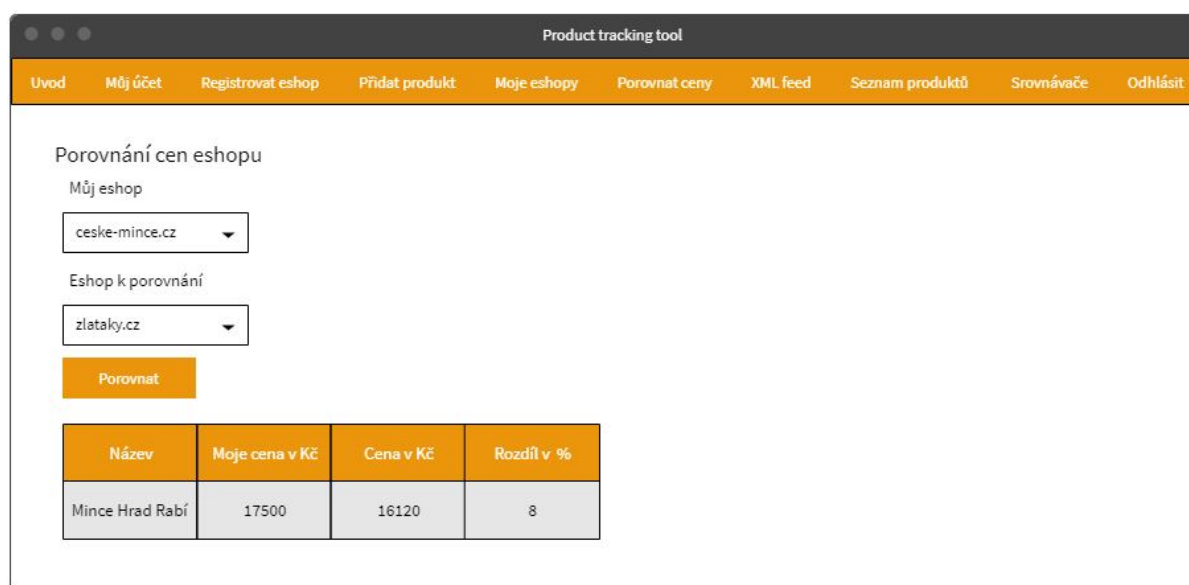
Obrázek 9: Registrace e-shopu

Po registraci e-shopu si může uživatel tento e-shop přidat do oblíbených a tím získat přístup k funkcím, které porovnávají ceny mezi e-shopy. Uživatel si vybere ze seznamu registrovaných a klikne na tlačítko "Registrovat". E-shopy, které má uživatel v seznamu oblíbených může kdykoliv ze seznamu odebrat.



Obrázek 10: Moje e-shopy

Pokud chce uživatel porovnat ceny mezi e-shopem, který má přidáný v oblíbených a libovolným jiným e-shopem, tak v menu aplikace vybere záložku "Porovnat ceny". Zde uživatel vybere jeden e-shop ze seznamu oblíbených a jeden ze seznamu registrovaných a klikne na tlačítko "Porovnat". Do tabulky se vypíše název porovnávaného produktu, cenu v oblíbeném e-shopu, cenu v porovnávaném e-shopu a rozdíl cen v procentech.



Obrázek 11: Porovnání cen

## Shrnutí

V první části této kapitoly jsem popsal využití technologie při implementaci nástroje pro sledování produktů, jejich důvod využití a upřednostnění před ostatními technologiemi. Také jsem popsal knihovny a balíčky, které jsem využil pro implementaci jednotlivých funkcí, uvedl jsem důvod jejich použití a případné existující alternativy, které by se daly pro implementaci využít. Ve druhé části jsem se věnoval návrhu uživatelského rozhraní. U každého návrhu jsem vysvětlil pracovní postup, jak s daným uživatelským rozhraním uživatel pracuje a uvedl model rozhraní.

## 6 Implementace

Tato kapitola bude věnována mému řešení implementace nástroje pro sledování produktů. Postupně bude popsán popis implementace jednotlivých částí nástroje, jako je například využití návrhových vzorů přenos dat mezi aplikací a databází, implementace návrhového vzoru MVC na platformě ASP.NET a implementaci uživatelského rozhraní a jeho porovnání s návrhem.

### 6.1 ASP.NET MVC

Nástroj je implementován na platformě ASP.NET v jazyce C#. Tato platforma implementuje návrhový vzor MVC. Ten rozděluje interakci uživatelského rozhraní do tří odlišných rolí. *Model*, který obsahuje nebo představuje data, se kterými uživatel pracuje, *View* pro vykreslení části modelu jako uživatelské rozhraní a *Controller*, který zpracovává příchozí požadavky, vykonává operace na modelu a vybírá, které view se má zobrazit [19]. V této implementaci model obsahuje *data transfer object* (viz. následující podkapitola) a veškeré třídy důležité pro zpracování všech formulářů, které se v aplikaci nacházejí. Část View obsahuje soubor s názvem *\_\_Layout*, který obsahuje společnou část pro všechny view soubory. V tomto případě se jedná o položky menu aplikace, kdy každý nepřihlášený a přihlášený uživatel má v menu stejné položky. Jednotlivé soubory view jsou rozděleny podle toho, který controller s nimi manipuluje. Controller, který slouží jako "prostředník" mezi view a modelem, získá například požadavek na zobrazení produktů z vybrané kategorie. Controller dostane požadavek na zobrazení produktů v kategorii se zadaným ID, z databáze načte produkty, vybere správný view a zašle uživateli odpověď. Takto v aplikacích ASP.NET funguje každý controller. Připojení k databázi a zasílání emailových zpráv uživatelům je implementováno pomocí vlastnosti ASP.NET Core, která se nazývá *Dependency Injection*. Díky DI controller nemusí znát, jak se požadovaný objekt vytváří. O vytváření DI objektů se stará service provider. Ten při vytváření nové instance controlleru zkontroluje jeho závislosti, vytvoří požadované instance a vloží je do controlleru [19]. Tak se může controller připojit k databázi a manipulovat s daty, nebo zasílat emailové zprávy.

---

```
public IActionResult Products(int pcID, string name)
{
    db.Connect();
    Collection<product> products = productGateway.SelectByProductCat(pcID, db);
    db.Close();
    ViewBag.products = products;
    ViewBag.name = name;
    return View();
}
```

---

Výpis 1: Ukázka akce controlleru Products

## 6.2 Datová vrstva

V datové vrstvě aplikace se nacházejí objekty, které se nazývají *data transfer object* [20]. Jedná se o objekty, které slouží k přenosu dat mezi procesy. Jednotlivé položky tohoto objektu jsou primitivní typy jako například *string*, *date* nebo jiný *data transfer object*. Jeden tento objekt představuje jednu tabulku, která se nachází v databázi. Máme tedy například DTO *user*, který odpovídá tabulce *uzivatel*. Převedení tabulky v relační databázi do objektu v objektově orientovaném programovacím jazyce se provádí pomocí objektově relačního mapování. ORM je programovací technika, která provede konverzi dat tak, aby byla dostupná pro objektové prostředí. Tím dojde k synchronizaci mezi používanými objekty v aplikaci a jejich reprezentací v databázi [21].

---

```
public class user
{
    public string jmeno { get; set; }
    public string prijmeni { get; set; }
    public string email { get; set; }
    public string heslo { get; set; }
    public int user_ID { get; set; }
}
```

---

Výpis 2: Ukázka DTO *user*

Pro přístup k databázi se používají *data access object* [22]. Tyto objekty zapouzdřují veškerý přístup k datovému zdroji. Pro jednu tabulku v databázi existuje jeden objekt DAO, který se stará o veškeré operace s daty v této tabulce. Metody v DAO vrací kolekci objektů DTO, která obsahuje výsledek dotazu v této metodě. Kromě metod, které zasílají na databázi požadavky pro vykonání SQL dotazů, obsahují DAO objekty také metody, které vykonávají uložené procedury a funkce v databázovém systému.

---

```
public static Collection<product> SelectByProductCat(int pcID, database db)
{
    string command = "select * from product where pcID = @pcID";
    SqlCommand com = new SqlCommand(command, db.connection);
    com.Parameters.AddWithValue("@pcID", pcID);
    SqlDataReader reader = com.ExecuteReader();
    Collection<product> products = new Collection<product>();
    while (reader.Read())
    {
        product p = new product();
```

---



```

        int i = -1;

        p.pID = reader.GetInt32(++i);
        p.eID = reader.GetInt32(++i);
        p.pcID = reader.GetInt32(++i);
        p.nazev = reader.GetString(++i);
        p.cena = reader.GetInt32(++i);
        p.dostupnost = reader.GetBoolean(++i);
        p.url = reader.GetString(++i);
        p.datum = reader.GetDateTime(++i);

        products.Add(p);
    }
    reader.Close();
    return products;
}

```

---

Výpis 3: Ukázka metody z DAO *product*

Všimněme si, že řetězec *command* v ukázce metody z DAO *product*, obsahuje parametrizovaný SQL dotaz. Díky tomu se zabrání tzv. Sql injection. Jedná se o napadení databázové vrstvy takovým způsobem, že útočník využije zranitelnosti neošetřeného SQL dotazu, který aplikace zasílá do databáze. Útočník tak může do vstupu aplikace zadat vlastní škodlivý SQL kód a napáchat škodu na datech nebo data odcizit [23].

### 6.3 Implementace uživatelského rozhraní

Uživatelské rozhraní v aplikacích ASP.NET Core MVC představuje část *View*. Ta obsahuje soubory s příponou *.cshtml*. Tyto soubory obsahují HTML kód, kód v jazyce C# pomocí Razor engine, nebo JavaScript. Razor engine slouží k dynamickému vytváření obsahu webových stránek a je zodpovědný za vkládání dat do HTML dokumentů [19]. Celé uživatelské rozhraní je tedy napsáno ve značkovacím jazyce HTML s dynamickou tvorbou stránek pomocí Razor engine a vykreslováním grafů pomocí JavaScript knihovny. Všechny webové stránky aplikace mají společné menu aplikace. Proto existuje soubor s názvem *\_\_Layout.cshtml*, který obsahuje základní strukturu HTML dokumentu a implementované menu aplikace. Proto každý *.cshtml* soubor nemusí obsahovat základní strukturu HTML dokumentu, ale pouze podstatný kód pro dané view. Grafický vzhled dokumentu je vytvořen pomocí kaskádových stylů CSS, které jsou upraveny tak, aby umožnily používání aplikace na mobilních zařízeních.

Název	Cenové rozpětí v Kč	Dostupných	Produkty	Sledovat	Aktualizovat
Mince Hrad Veverí (3)	1000 - 0	0	Produkty	Přidat ke sledování	Aktualizovat
Mince 100. výročí Vznik Československa (3)	46000 - 44900	2	Produkty	Přidat ke sledování	Aktualizovat
Zlatá investiční mince Maple Leaf 1 Oz (1)	31093 - 31093	1	Produkty	Přidat ke sledování	Aktualizovat
Mince Hrad Zvíkov (4)	17900 - 16350	3	Produkty	Přidat ke sledování	Aktualizovat
10000 Kč Zlatá bula sicilská 2012 Standard (3)	47800 - 37900	1	Produkty	Přidat ke sledování	Aktualizovat
Zlatá investiční mince Kangaroo Klokán 1 Oz (1)	31184 - 31184	1	Produkty	Přidat ke sledování	Aktualizovat
Mince Hrad Rábí (4)	17500 - 15900	2	Produkty	Přidat ke sledování	Aktualizovat
Hrad Bouzov (4)	19950 - 17800	3	Produkty	Přidat ke sledování	Aktualizovat
Hrad Pernštejn (4)	18500 - 16250	3	Produkty	Přidat ke sledování	Aktualizovat
Hrad Kost (6)	22900 - 18300	5	Produkty	Přidat ke sledování	Aktualizovat
Mince 100. výročí Vznik Československé měny (1)	39590 - 39590	0	Produkty	Přidat ke sledování	Aktualizovat
Negrelliho viadukt v Praze (2)	19900 - 18299	2	Produkty	Přidat ke sledování	Aktualizovat
Mince Národní kulturní památka papírna Velké Losiny (1)	66800 - 66800	1	Produkty	Přidat ke sledování	Aktualizovat
Hrad Bezděz (2)	22900 - 19000	1	Produkty	Přidat ke sledování	Aktualizovat

Obrázek 12: Screenshot uživatelského rozhraní aplikace

## Shrnutí

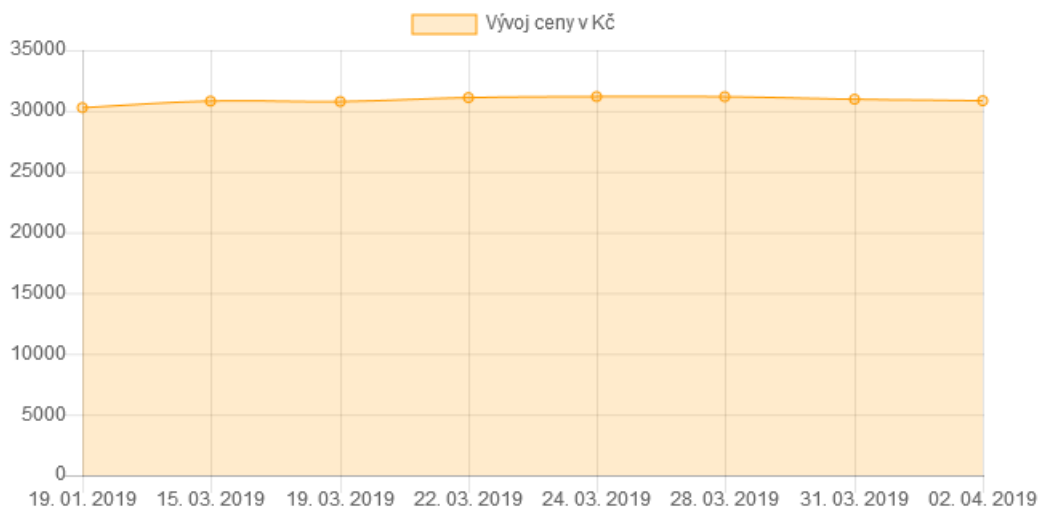
Tato kapitola byla věnována implementaci nástroje pro sledování produktů. V první části této kapitoly je popsán postup implementace webové aplikace pomocí platformy ASP.NET Core MVC. Byl vysvětlen význam návrhového vzoru *Model-View-Controller* a jeho částí v implementaci tohoto nástroje. Dále byl vysvětlen význam *dependency injection* a jeho využití při implementaci databázového připojení. V druhé podkapitole byla objasněna implementace datové vrstvy aplikace. Zjistili jsme, že pro implementaci přístupu k databázi je využito návrhového vzoru *data access object* a pro přenos dat mezi databází a aplikací slouží objekty *data transfer object*, které byly vytvořeny pomocí objektově relačního mapování. K oběma návrhovým vzorům byla uvedena ukázka zdrojového kódu. Poslední podkapitola této části byla věnována implementaci uživatelského rozhraní, které bylo implementováno pomocí značkovacího jazyka HTML a kódu v jazyce C# pomocí Razor engine. Dále byl vysvětlen význam souboru *\_\_Layout.cshtml* a způsob realizace grafické podoby webových stránek.

## 7 Testování

Tato kapitola je věnována výsledkům třítydenního testování aplikace. Během tohoto časového období bylo pomocí tohoto nástroje sledováno několik desítek náhodných produktů ze čtyř vybraných internetových obchodů. Tyto produkty byly pravidelně aktualizovány pomocí funkce nástroje pro aktualizaci dat produktů. V rámci testovacího procesu s aplikací také pracoval nezávislý uživatel, který poskytl zpětnou vazbu o jeho práci s nástrojem a podněty pro zlepšení implementace nástroje.

### Testování funkčnosti nástroje

Pro testování funkčnosti nástroje bylo vybráno několik desítek produktů celkem ze čtyř internetových obchodů se sběratelskými mincemi. Pomocí tohoto nástroje a jeho funkcí byl sledován vývoj jejich ceny a dostupnosti během tří týdnů. Každé dva až tři dny byla stažena nová data o produktech pomocí funkce nástroje pro aktualizaci produktu (viz strana 31). Při aktualizaci dat nedocházelo k žádným problémům a pro každý produkt byly získány aktuální informace o ceně a dostupnosti. Vývoj ceny a dostupnosti produktů jde vidět pomocí funkce pro zobrazení historie produktů. Při testování aplikace šlo vypořádat, že zejména produkty *Zlatá investiční mince Maple Leaf 1 Oz* a *Zlatá investiční mince Kangaroo Klokán 1 Oz* měnily svou cenu (viz obrázek 13). Na emailovou adresu, která byla použita pro testovací účet, přicházely emailové zprávy s upozorněním o změně ceny produktu, na který bylo nastaveno upozornění. Funkce, která má tyto zprávy posílat, funguje správně.



Obrázek 13: Graf vývoje ceny produktu *Zlatá investiční mince Maple Leaf 1 Oz*

Veškerá správa sledovaných produktů, nastavených upozornění a oblíbených e-shopů fungovala během testování správně bez jakýchkoliv problémů. Při testování funkce pro porovnání cen

e-shopů aplikace spadla, když jeden z porovnávaných produktů měl nulovou cenu, jelikož ho daný e-shop přestal prodávat. Tato chyba byla včas opravena a nyní již funkce pracuje správně.

### **Testování nástroje uživatelem**

Nástroj byl testován uživatelem, který při práci postupoval jako zcela nový uživatel aplikace. První věc, kterou uživatel otestoval, byly formuláře pro registraci a přihlášení. Při registraci ani při přihlášení do aplikace se nevyskytly žádné potíže a veškerá nastavená omezení fungovala správně. Dále uživatel testoval funkci pro registraci nového e-shopu. Při tomto testování uživatel objevil vážné nedostatky v registračním formuláři a nepodařilo se mu do nástroje přidat nový e-shop. Tyto nedostatky již byly na základě zpětné vazby opraveny a uživatelé by neměli mít s registrací nového e-shopu vážné problémy. Další podnět pro zlepšení nástroje, který uživatel navrhl, je zlepšení uživatelského rozhraní, které ohodnotil jaké nepřehledné. Uživatel při testování dalších funkcí neobjevil žádné další problémy a jednotlivé funkce nástroje a jeho účel pochopil.

### **Shrnutí**

Tato kapitola byla věnována testování funkčnosti implementovaného nástroje pro sledování produktů. V první části testování jsem se věnoval třítydennímu sledování vývoje cen a dostupnosti náhodných produktů ze čtyř e-shopů a k testování veškerých funkcí nástroje. Při tomto testování byla objevena a také opravena chyba ve funkci porovnání cen e-shopů. Ve druhé části došlo k testování nástroje uživatelem. Uživatel po testování poskytl zpětnou vazbu, na základě které byl opraven formulář pro registraci nového e-shopu a při opětovném testování již nedělala registrace e-shopu uživateli potíže. Ostatní podněty ze zpětné vazby uživatele byly sepsány v následující kapitole.

## 8 Návrhy na zlepšení

V této kapitole se zaměřím na možnosti zlepšení nástroje pro sledování produktů, který jsem v rámci této bakalářské práce implementoval. Tento typ nástrojů má velké množství dostupných možností, jak se získanými daty o produktech pracovat a poskytnout tak uživateli co nejlepší služby. Řada již existujících nástrojů obsahuje spoustu užitečných funkcí, které uživatelé hojně využívají, ale obsahují také spoustu funkcí, které skoro nikdo nevyužívá. Rád bych tedy v této kapitole vybral a popsal ty funkce, které by byly dobrým doplňkem k již stávajícím funkcím tohoto nástroje. Také bych zde napsal návrhy na zlepšení již stávajících funkcí a implementace nástroje.

### Export dat do XML a CSV

První funkcí, která by byla dobrým doplňkem k stávajícím funkcím nástroje, je export sledovaných produktů do souborů ve formátu XML a CSV. Funkce by fungovala následujícím způsobem. Uživatel by měl u svého seznamu sledovaných produktů možnost "Exportovat do XML" nebo "Exportovat do CSV". Po zvolení této možnosti by se veškerá data za poslední měsíc o produktech, které uživatel sleduje, uložila do souboru ve zvoleném formátu a stáhla se do uživatelova počítače. Takhle by měl uživatel možnost s informacemi o svých sledovaných produktech pracovat i mimo tento nástroj a mohl by je zpracovat jinou aplikací nebo informačním systémem, archivovat tyto data nebo je například sdílet s obchodními partnery.

### Měsíční report

Další funkcí, která by zlepšila poskytované služby tímto nástrojem pro sledování produktů je zasílání pravidelných reportů ve formátu *.xlsx* jednou za měsíc. Tento report by obsahoval údaje o sledovaných produktech, grafy vývoje jejich ceny, porovnání cen v e-shopu uživatele s cenami konkurence, nebo například výpis produktů, které prodává konkurence, ale uživateli v jeho obchodě chybí. Díky této funkci by měl uživatel na jednom místě přehled o stavu sledovaných produktů, cenovém srovnání jeho produktů s konkurencí a nápady na nové produkty do jeho obchodu, které by mohly zajistit vysoký zisk.

### Známkování produktů

Tato funkce nástroje by pomocí algoritmu dávala jednotlivým produktům hodnocení na škále 1 až 5, kdy známka 1 by byla nejhorší a známka 5 nejlepší. Tento algoritmus by ohodnotil produkt na základě vývoje jeho ceny a dostupnosti a velikosti konkurence. Například produkt, jehož cena by rostla a byl by dostupný, ale existovala by spousta obchodů, které by tento produkt prodávaly, tak by tento produkt dostal hodnocení 4. S klesající konkurencí by se naopak hodnocení tohoto produktu zvyšovalo k hodnotě 5. Celkové hodnocení produktu by znamenalo, jestli by byl produkt pro uživatele dobrou investicí nebo nikoliv.

## **Zobrazování recenzí**

Pomocí této funkce by si mohl uživatel zobrazit hodnocení produktů zákazníky a tím určit, který produkt je mezi zákazníky oblíbený a vyplatí se do něj investovat. Funkce by daná hodnocení získávala podobným způsobem, jako to dělají ostatní funkce tohoto nástroje, které získávají data o produktech (viz funkční analýza). Kromě recenzí produktů by si mohl uživatel také zobrazit hodnocení svého e-shopu a zjistit, jak jeho e-shop hodnotí zákazníci, zjistit své nedostatky a napravit je. Jednalo by se tedy o funkci, která by pomohla uživateli s vylepšením poskytovaných služeb jeho e-shopu na základě recenzí zákazníků a s rozhodnutím, který produkt přidat do sortimentu svého e-shopu.

## **Administrátorské rozhraní**

Hlavní funkce, která v aplikaci chybí, je administrátorské rozhraní aplikace. Veškerá správa uživatelů, produktů a e-shopů v momentálním stavu aplikace probíhá pomocí Microsoft SQL Server Management Studio, tudíž je pro správu potřeba správce, který má znalosti v dotazovacím jazyce SQL. Vytvoření administrátorského rozhraní by tuto potřebu eliminovalo a poskytlo by přehlednější a jednodušší správu aplikace a kvalitnější poskytnutí uživatelské podpory.

## **Uživatelské rozhraní**

Na základě zpětné vazby v rámci testování aplikace by bylo potřeba vylepšit uživatelské rozhraní aplikace, které se pro uživatele zdá být nepřehledné a lehce se v něm ztrácí. To může vést k nepohodlné práci s nástrojem a k potřebě kontaktovat uživatelskou podporu. Jako řešení by bylo potřeba zlepšit dokument s CSS styly tak, aby uživatelské rozhraní bylo pro uživatele přehlednější a příjemnější na pohled a práce s ním byla intuitivnější.

## **Shrnutí**

V této kapitole jsem popsal možné návrhy na zlepšení tohoto nástroje pro sledování produktů zejména z hlediska přidání nových funkcí, které by pomohly uživatelům lépe sledovat produkty a zpracovat data. Kromě nových funkcí jsem také zmínil možnost implementace administrátorského rozhraní pro zlepšení uživatelské podpory a jednodušší a přehlednější správy nástroje. Jako poslední jsem zmínil vylepšení formuláře pro registraci nového e-shopu na základě výsledků testování aplikace.

## 9 Závěr

Z analýzy existujících nástrojů pro sledování produktů lze vypožorovat, že každý z těchto nástrojů porovnává ceny produktů uživatele s produkty, které prodávají konkurenční e-shopy a uživatelé. Zároveň uživatelům poskytují velkou databázi dat o produktech, kterou mohou uživatelé procházet a sledovat, které produkty prodávají konkurenční prodejci a zároveň sledovat cenový vývoj jednotlivých produktů. Ostatní poskytované služby a funkce nástrojů se od sebe liší, ale některé funkce fungují podobným způsobem.

V praktické části bakalářské práce jsem implementoval své vlastní řešení nástroje pro sledování produktů. Cílem této práce bylo implementovat nástroj, který by uživatelům dovolil registrovat svůj vlastní e-shop a sledovat vývoj ceny a dostupnosti produktů z těchto registrovaných e-shopů. Na základě výsledků testování nástroje lze usoudit, že bylo těchto cílů dosaženo a nástroj splnil svůj účel.

Implementovaný nástroj obsahuje v porovnání s existujícími nástroji pro sledování produktů velmi podobné funkce, ale žádné z existujících řešení neobsahuje možnost registrace e-shopu a přidání nového produktu samotným uživatelem, nebo sledování produktů na inzertních portálech.

## Literatura

- [1] Jungle Scout Web App [online]. [cit. 2019-03-22]. Dostupné z: <https://www.junglescout.com/the-web-app/>
- [2] Jungle Scout Chrome Extension [online]. [cit. 2019-03-22]. Dostupné z: <https://www.junglescout.com/the-chrome-extension/>
- [3] Jungle Scout Review [online]. [cit. 2019-03-22]. Dostupné z: [https://www.trustpilot.com/review/junglescout.com?utm\\_medium=Trustbox&utm\\_source=Carousel](https://www.trustpilot.com/review/junglescout.com?utm_medium=Trustbox&utm_source=Carousel)
- [4] Unicorn Smasher [online]. [cit. 2019-03-22]. Dostupné z: <https://www.unicornsmaasher.com/#>
- [5] Unicorn Smasher Reviews, Pricing & Popular Alternatives [online]. [cit. 2019-03-22]. Dostupné z: <https://fitsmallbusiness.com/unicorn-smasher-reviews-pricing/>
- [6] AMZ Tracker [online]. [cit. 2019-03-22]. Dostupné z: [https://www.amztracker.com/index\\_en.php](https://www.amztracker.com/index_en.php)
- [7] AMZ Tracker Reviews, Pricing & Popular Alternatives [online]. [cit. 2019-03-22]. Dostupné z: <https://fitsmallbusiness.com/amz-tracker-reviews-pricing/>
- [8] AMZ Tracker Review [online]. [cit. 2019-03-22]. Dostupné z: <https://www.amzfinder.com/blog/amz-tracker-review-best-one-amazon-seller-tool/>
- [9] About Cheap Shark [online]. [cit. 2019-03-22]. Dostupné z: <https://www.cheapshark.com/about>
- [10] Price Check [online]. [cit. 2019-03-22]. Dostupné z: <https://www.price-check.cz/>
- [11] Price Check pro eshopy [online]. [cit. 2019-03-22]. Dostupné z: <https://www.price-check.cz/pro-eshopy>
- [12] Price Check pro vendory [online]. [cit. 2019-03-22]. Dostupné z: <https://www.price-check.cz/pro-vendory>
- [13] Funkce Price Check [online]. [cit. 2019-03-22]. Dostupné z: <https://www.price-check.cz/funkce>
- [14] Dataweps - Azor [online]. [cit. 2019-03-22]. Dostupné z: <https://www.dataweps.com/cs/azor/>
- [15] Dataweps - Beed [online]. [cit. 2019-03-22]. Dostupné z: <https://www.dataweps.com/cs/beed/>



- [16] Dataweeps - Disivo [online]. [cit. 2019-03-22]. Dostupné z: <https://www.dataweeps.com/cs/disivo/>
- [17] Dataweeps - SocialWatch [online]. [cit. 2019-03-22]. Dostupné z: <https://www.dataweeps.com/cs/socialwatch/>
- [18] Dataweeps - TrendLucid [online]. [cit. 2019-03-22]. Dostupné z: <https://www.dataweeps.com/cs/trendlucid/>
- [19] FREEMAN, Adam. Pro ASP.NET Core MVC. Sixth Edition. New York, NY: Springer Science Business Media, 2016. ISBN 978-1-4842-0398-9.
- [20] FOWLER, Martin. Patterns of enterprise application architecture. Boston: Addison-Wesley, c2003. ISBN 03-211-2742-0.
- [21] Objektově relační mapování [online]. [cit. 2019-03-23]. Dostupné z: [https://cs.wikipedia.org/wiki/Objektov%C4%9B\\_rela%C4%8Dn%C3%AD\\_mapov%C3%A1n%C3%AD](https://cs.wikipedia.org/wiki/Objektov%C4%9B_rela%C4%8Dn%C3%AD_mapov%C3%A1n%C3%AD)
- [22] ALUR, Deepak, Dan MALKS a John CRUPI. Core J2EE patterns: best practices and design strategies. Enterprise edition. Upper Saddle River: Prentice Hall, c2001. ISBN 01-306-4884-1.
- [23] CLARKE, Justin. SQL injection attacks and defense. Second Edition. Burlington, MA: Syngress Pub., c2009. ISBN 15-974-9424-0.